Implementation of a Slovene Language-Based Free-Text Retrieval System

A study submitted in fulfilment of
the requirements for the degree of
Doctor of Philosophy
at
The University of Sheffield

by

Mirko Popovič

Department of Information Studies

June 1991

Mirko Popovič June 1991

Implementation of a Slovene Language-Based Free-Text Retrieval System

Abstract

This thesis is concerned with providing end-user access to bibliographic databases in Slovenia. A statistically-based approach to document retrieval, in particular nearest neighbour searching, is selected as a means to achieve this goal. The following two main questions are investigated in the context of this thesis: (a) are statistically-based techniques applicable to Slovene information retrieval systems, and (b) could statistically-based techniques provide a framework for developing a multi-lingual information retrieval system?

After providing a theoretical background to the experimental work, the design of a stop-word list and a stemming algorithm for Slovene is discussed. The resulting stop-word list contains a total of 1,593 non-content bearing words. Two stemming algorithms are described, one context-free and the other context-sensitive; the latter is found to be far more effective in operation, owing to the large number of context sensitive and recoding rules that are required to reflect fully the morphology of Slovene.

The retrieval effectiveness of this stemming algorithm is evaluated within the best-match context, using the Slovene version of the INSTRUCT package. The performance of the stemming algorithm is tested by its comparison with two other types of text representation, i.e., manual right-hand truncation carried out by a trained intermediary, and unstemmed words. The results of this comparative evaluation reveal the following: (a) there is a significant performance difference between automatic word conflation and unstemmed processing of the Slovene text; (b) there is no significant performance difference between automatic stemming and manual right-hand truncation, carried out by a trained intermediary. It follows that one of the important components of an information retrieval system, i.e., word conflation, can be automated in Slovene systems with no average loss of performance, thus allowing users easier access to the systems.

Having obtained good performance results with the employment of the Slovene stemming algorithm, a multi-lingual experiment is described. Its main objective is to test the performance of statistically-based techniques in two different languages, i.e., Slovene and English. A detailed analysis of performance results confirms only one of three main hypotheses. Although the experiment on the identification of stem variants produces a similar number of related terms from both English and Slovene dictionary components of the inverted file, the other two hypotheses are rejected, i.e., (a) processing of the English documents and queries does not produce more or less identical hits to those retrieved from the Slovene database; (b) the Slovene version of INSTRUCT produces significantly better performance results than its English equivalent. The employment of a failure analysis reveals two main causes of performance difference, i.e., the frequent occurrence of synonyms and other related terms, and the automatic word conflation carried out by the two different stemming algorithms.

On this basis, conclusions and suggestions for further work are given. It is emphasized that advanced, statistically-based techniques of information retrieval will be firmly established in Slovenia only if they can be enhanced with refinements which allow a multi-lingual approach to document retrieval.

Acknowledgements

I would like to give my sincere thanks to all those who helped me during the course of this PhD project: in particular, Dr Peter Willett, my supervisor, for his encouragement, guidance, and professionalism.

Unfortunately, the participants who gave so much of their valuable time to evaluation are too numerous to mention, but this project would not have been possible without them. However, I would like to emphasize a work carried out by Boris Košorok, a professional intermediary at the National & University Library in Ljubljana; without his help, it would have been impossible to complete all the experimental tests.

Sincere thanks are offered also to Richard Gilbert, of the Computing Centre at the University of Sheffield, who gave me valuable information on how to use international e-mail facilities. His knowledge was of a particular importance in the second phase of my research, while I was working in Ljubljana, and frequent communication with Sheffield became a rule.

Since English is not my native language, I would like to thank my colleagues Janey Cringean, Val Gillett, and Helen Grindley for their time and effort in clarifying some of the expressions in my thesis. In particular, I would like to show gratitude to Helen Grindley for taking the time to read the final draft of my thesis and for providing useful comments.

Thanks are owed also to the British Council, the Ministry of Culture in Slovenia, the Ministry of Research and Technology in Slovenia, and the National & University Library in Ljubljana, for their financial support.

Finally, I would like to thank my wife, Breda, for her encouragement, time and patience. This PhD thesis is Breda's as much as it is mine.

Contents

Preface

The provision of end-user searching facilities has been recognized as the only way to remove a barrier between the original source of a query and the query's answer. This thesis is therefore aimed at increasing the possibilities of easy end-user access to bibliographic databases in Slovenia. At present, end-users in Slovenia are faced not only with a growing number of bibliographic and other types of databases, but also with a multi-lingual information retrieval environment. In other words, they are surrounded by document collections, written in many different languages (i.e., Slovene and other Yugoslav languages, major European languages). In addition, all software systems (e.g, ATLASS, TRIP) available for accessing these databases are typical of current retrieval software elsewhere in that they are based on Boolean searching, with professional intermediaries being used to carry out on-line searches on behalf of end-users. Consequently, modern, non-conventional methods and techniques of information retrieval which allow direct, end-user interaction with the system are neither incorporated into existing retrieval systems in Slovenia, nor has much research been carried out in this area.

One of the main research areas in information retrieval is the development of algorithmic procedures which allow the computer to undertake many of the functions of a trained intermediary. This approach, based on the use of a range of statistical techniques—also known as the *statistically-based* approach to document retrieval—has been used in this thesis to develop a Slovene language-based free-text retrieval system. Therefore, the main problem which was investigated in the context of this PhD project, is contained in the following two questions:

1. Are statistically-based techniques applicable to Slovene information retrieval systems?

2. Could statistically-based techniques provide a framework for developing multilingual information retrieval systems?

The thesis begins with an introduction and description of recent trends in document retrieval (Chapter 1). Particular attention is given to the statistically-based approach

to information retrieval which is based on the following main components: automatic indexing, nearest neighbour searching, and term weighting. Most of the statistically-based techniques are independent of a particular language, with one exception, i.e., processing by a stemming algorithm. Chapter 2 therefore contains a detailed review of the automatic word conflation techniques which can be used to increase the effectiveness and efficiency of information retrieval systems. The development of a Slovene language-based free-text retrieval system required the design of an effective stemming algorithm for the Slovene text. Since such a design must take into account the language's morphological structure, Chapter 3 presents the main characteristics of contemporary Slovene, with particular reference to its inflectional morphology. The design of a stop-word list and a context-sensitive stemming algorithm for Slovene—together with their evaluation—are covered in Chapter 4. However, the retrieval effectiveness of these two language-dependent procedures cannot be evaluated without their incorporation into a retrieval system. The INSTRUCT package was used as a test bed for this experiment and is thus outlined in Chapter 5, together with a description of a Slovene version of INSTRUCT which required the implementation of language-dependent procedures. Chapter 6 discusses a test environment (a test collection, text representation modules) which was built to evaluate the effectiveness of the Slovene stemming algorithm. The performance of this algorithm was compared with the two other types of text representation, i.e., manual right-hand truncation and non-conflation. Chapter 7 deals with results obtained from this experiment. Chapter 8 compares the performance of the Slovene information retrieval system with its English equivalent in order to find out whether statistically-based techniques—as implemented in INSTRUCT—could provide a framework for a multi-lingual approach to document retrieval. Finally, Chapter 9 presents the conclusions and suggests areas for future work.

Appendices A–D contain a list of literature which was consulted to describe the main characteristics of the Slovene language, and the three sets of queries that were used. A floppy disk contains the lists of stop-words and suffixes that were identified during the project.

Chapter 1

Recent Trends In Document Retrieval

1.1 Introduction

1.1.1 Document retrieval – a definition

The term *document retrieval* can best be described within the general framework of information retrieval which is concerned with the representation, storage, organization, and accessing of information items (Salton and McGill, 1983). The term information retrieval covers a wide range of disciplines with the emphasis on non-numeric computing (e.g., document retrieval, natural language processing, database management systems).

Historically, however, the term information retrieval (IR) has been used to cover the storage, processing and retrieval of information from databases containing bibliographical details on documents of all kinds (e.g., books, journal articles, reports, patents). In these databases, the input information consists of natural language text (e.g., bibliographic citations, abstracts), and the output to a search request comprises sets of references. These references are intended to provide end-users with information about items of potential interest. Thus, information retrieval systems can alternatively be described as document retrieval systems, or even more precisely, as *reference retrieval*

systems (Willett, 1988a). The main task of these systems is to (Salton and McGill, 1983):

- retrieve documents and references;
- store natural language texts; and
- process users' queries.

However, the rapid spread of full-text databases (Tenopir, 1984) as a result of changing technology (e.g., the trend towards electronic publishing, use of optical readers to scan text) also leads to the application of document retrieval techniques to textual materials themselves.

1.1.2 Some characteristics of current document retrieval systems

In today's changing society in which information and knowledge play a crucial role in socioeconomic development there is increasing interest in the use of document retrieval systems. The current trends can be summarized in Goldsmith's words (1982):

> "... computerized text retrieval has been a concept and a reality for some years but has not really been a workable proposition until recently, probably because of lack of sufficient computer power and because of a general lack of awareness, and hence commitment, by those most likely to benefit" (p. 41).

Computerized information retrieval systems have now been in use for almost thirty years, starting with the pioneering work in KWIC indexing, followed by the development of SDI systems, current awareness and retrospective searching. However, it is technological progress, i.e., increases in computer processing power and capacity, improvements in telecommunications, reductions in the cost of direct access storage devices—together with increasing user demand for accurate and up-to-date information—which has led to a growing number of both the large on-line, external databases, and internal, so-called *in-house* databases. In addition to a growing number of on-line bibliographic

databases (Williams, 1985), there is increasing interest in other types of data, and thus different kinds of databases (e.g., numerical databanks, full-text databases).

To make efficient and effective use of textual data in both external and internal databases sophisticated methodologies and techniques are needed to store, process, and transmit information. Surprisingly, despite great technological developments, it is often claimed that current information retrieval systems are unable to deal effectively with the ever increasing growth of information and that current operational capabilities remain at a relatively elementary stage (Salton and McGill, 1983).

Specifically, there are two essential features in existing document retrieval systems:

- inverted file organization;
- use of Boolean operators.

In addition to the source data file, inverted file organization consists of the following two main files: the dictionary file, and the postings file. In the *dictionary file*, indexing terms are listed in alphabetical order along with the total number of documents in which they occur. The *postings file* lists the indexing terms with the accession number of each of the documents in which a term appears. This file is used to "point" to the documents in the main sequential *data file* so that the full record can be retrieved. Taken together, the dictionary and postings files are generally referred to as an *inverted file*.

Although inverted file organization requires additional storage and maintenance, its crucial advantage over serial file organization is its very rapid response to queries. Instead of inspecting each document in a database, only a few documents need to be matched against a query because the data file is inverted to provide indexes to documents containing the query terms.

Such a file organization, using individual terms and document reference numbers, can handle Boolean operators particularly easily by translating AND, OR, and NOT operations into set intersection, set union, and set difference, respectively. Consequently, Boolean searching techniques—with some additional facilities, e.g., truncation

and proximity searching—dominate in the present document retrieval systems.

Why, then is it so often reported in the literature that these systems are not capable of entirely meeting users' needs, and, moreover, that they are hostile to end-users (Cleverdon, 1984)? It is stressed by Pollitt (1986) that,

> "...computerised searching services will not have their full impact upon user communities until direct user searching is widespread" (p. 1).

One of the consequences of the increasing number of large databases is the appearance of so-called *professional intermediaries* who are required to carry out on-line searches on behalf of end-users. These people are needed to help formulate user requests and to provide guidance on how the system is organized, on what materials are available, and how to search for and locate the desired items. This is a very questionable situation because it is very difficult to determine what the user really requires and the intermediary may seldom be aware of a user's real needs. It is widely believed that end-users will be able to make their own requests on-line when search processes are simplified or made more "friendly".

A need for professional intermediaries is recognized as a limitation of current information retrieval systems since either end-users are excluded from the search process or extensive training is required to carry out searches in a cost-effective manner. It is claimed by Cleverdon (1984) that,

> "...the error rather lies in a failure to exploit correctly the resources of the computer, due to a lack of understanding of the nature of retrieval systems" (p. 38).

At present there are two main research areas in information retrieval whose main aim is to enable end-users to carry out searching in both an efficient and effective manner. One of these areas is research into expert intermediary systems to provide intelligent front-ends to bibliographic databases. This approach, also known as a *knowledge-based* approach to information retrieval systems, uses expert systems techniques (e.g., rule-based programming) to encode the expertise possessed by a trained intermediary. This research has resulted in some operational systems, for example CONIT (Marcus,

1983), and CANSEARCH (Pollitt, 1986), which enable end-users to undertake searches without the knowledge or training of a professional intermediary. The main advantage of these systems is communication with the database via the user's natural language, the main drawback a domain dependence.

A quite different area of research in enabling on-line searching to be carried out by end-users is based on the development of algorithmic procedures which allow the computer to undertake many of the functions of a trained intermediary. This approach, based on the use of a range of statistical techniques, is also known as the *statistically-based* approach to information retrieval. Research in this area which is also concerned with retrieval effectiveness, i.e., to retrieve a larger amount of relevant material than Boolean systems, has resulted in some operational systems, e.g., SMART, SIRE (Salton and McGill, 1983), MASQUERADE (Brzozowski, 1983), CUPID (Porter, 1982), CITE (Doszkocs, 1983), INSTRUCT (Hendry *et al.*, 1986a), and MUSCAT (Porter and Galpin, 1988). These, and some other systems—although in their early stages—have clearly demonstrated many advantages over conventional, Boolean text retrieval systems. At the heart of this new generation of free-text retrieval systems are the following techniques:

- automatic indexing;
- best-match retrieval;
- term weighting.

There is no doubt that with developments in hardware and software technology, together with the trend of increasing use of textual databases, both areas of research will become very attractive as an alternative to conventional, Boolean information retrieval. This has already been confirmed by the implementation of non-conventional information retrieval techniques in commercial retrieval systems, as for example STATUS/IQ (Pape and Jones, 1988). Moreover, there are some indicators (Wade *et al.*, 1988) that the best results in information retrieval can be achieved by the integration of knowledge-based and statistically-based retrieval techniques. However, tests on a much

larger scale are needed to identify the optimum way of integrating the two approaches.

Therefore, this chapter will describe the statistically-based approach to information retrieval which has been in existence for more than two decades and which is beginning to show successful results in various operational systems. In the first section, a comparison between manual and automatic indexing will be outlined, followed by a description of automatic term selection and word conflation. The second section will be concerned with best-match searching as an alternative to Boolean searching. In the third section the main emphasis will be on describing search term weighting models. In the last section, some other areas of advanced research in information retrieval will also be briefly outlined.

1.2 Automatic indexing

1.2.1 Comparison between manual and automatic indexing

It is generally accepted that of all the operations required in information retrieval, the most crucial and probably the most difficult one consists of assigning appropriate terms and identifiers capable of representing the content of documents in a particular database. These terms, known also as descriptors, indexing terms, or keywords act as secondary keys for the retrieval of those documents which appear most similar to the given query formulations.

Indexing can be performed either manually by trained experts or automatically by computers. The first predictions about the future of content analysis tasks were in favour of manual indexing:

> "It is very likely that manual indexing (content analysis) by cheap clerical labor will still, on average, be qualitatively superior to any kind of automatic indexing. ... neither the assignment of topic terms to a given request, nor the reformulation of a request are processes which could conceivably be adequately mechanized, contrary to some speculation in this direction" (Bar Hillel, 1962; cf. Salton, 1986, p. 1).

Although the perception about "cheap clerical labour" has drastically changed since

such predictions, indexing is still been carried out manually by trained indexers. To this day, manual indexing is the rule rather than the exception in most operational environments.

In many manual indexing situations, where trained personnel are involved, the use of a controlled indexing language is preferred (i.e., a single standard term or phrase represents a wide variety of related terms and descriptions). This means that a variety of aids are made available to the indexer to control the indexing process, including for example a thesaurus that contains lists of equivalent and related terms for each standard thesaurus entry, or hierarchical dictionaries that contain general term arrangements capable of identifying broader and narrower terms for the various dictionary entries.

In addition, to obtain quality, accuracy, and consistency of performance in manual indexing, considerable demands are placed on the indexing personnel. It is expected that trained indexers should not only be knowledgeable about the subject matter of the database, but should also be familiar with the available indexing vocabularies and practices (e.g., number of terms, thesaural relationships). Furthermore, the performance of the various indexers should be sufficiently consistent to guarantee that similar documents are identified by comparable indexing entries. Thus, a great deal of training, experience and knowledge is required from trained indexers.

If the above demands are met, it is possible in principle to generate very useful manual indexing products. However, as said by Salton (1986), the practice of manual indexing is often different from the theory. The results of an investigation into various aspects of the storage and retrieval process (Cleverdon, 1984) indicated that:

- if two people or groups of people construct a thesaurus in a given subject area, only 60% of the index terms may be common to both thesauri;
- if two experienced indexers index a given document using a given thesaurus, only 30% of the index terms may be common to the two sets of terms;
- if two intermediaries search the same question on the same database on the same host, only 40% of the output may be common to both searches;

- if two scientists or engineers are asked to judge the relevance of a given set of documents to a given question, the area of agreement may not exceed 60%.

Thus, even if the indexing process were carried out accurately, and at the right level of detail, it is actually impossible to perform the indexing procedure consistently since more than one indexer will necessarily be needed in practice. This inevitable inconsistency affects retrieval performance and therefore leads to doubts about the potential advantages of strictly controlled, manually applied indexing languages.

The disadvantages and limitations of manual indexing have led to an increasing interest in an alternative approach to content analysis, i.e., in *automatic indexing*, where the selection of content identifiers is carried out with the aid of computing equipment. In automatic indexing, the problems caused by the use of a controlled language thesaurus and variations in indexing are avoided; in addition, there is also the possibility of natural-language searching in a document collection.

Research into automatic indexing has resulted in a wide range of techniques, which have been implemented in either experimental or operational retrieval systems. Tests on these systems have shown that simple automatic indexing methods are fast and inexpensive, and produce a *recall* (i.e., proportion of relevant material retrieved) and *precision* (i.e., proportion of retrieved material actually relevant) performance at least equivalent to that obtainable in a manual, controlled term environment (Salton and McGill, 1983).

Results of research work have also led, according to Willett (1988a), to a general agreement that an automatic indexing system should consist of the following components:

- a term selection module, which is responsible for the selection of descriptors on the basis of a text analysis of a document;
- a conflation procedure, which is used to reduce variants of a word to a single canonical form;
- a weighting mechanism, which assigns measures of relative importance to the

words which have been selected as document identifiers.

At this point it is necessary to emphasize that a term weighting mechanism can be used either for the automatic selection of indexing terms or for weighting of query terms at search time. Since more useful results have been obtained by the implementation of different weighting schemes in the latter approach, a term weighting mechanism will be outlined in connection with best-match searching. Thus, only a term selection module and a conflation procedure will be described in this section.

1.2.2 Statistical approach to automatic term selection

It has already been said that the indexing task consists of assigning to each stored item terms capable of representing document content. The first and most obvious place where appropriate descriptors might be found is the text of the documents themselves, or the text of document titles and abstracts. Thus, this section is concerned with methods for the automatic extraction of content terms from documents and document excerpts.

In automatic indexing, there have historically been two main approaches to term selection:

- linguistic approach, based on semantic and syntactic theories of automatic indexing;
- statistical approach, based on the concept of word frequencies.

Linguistic techniques, very popular in the 1960s, have not generally met expectations in automatic indexing research (Willett, 1988a). Although they were seen as quite a desirable computing technique, their integration into information retrieval did not prove to be as easy as was initially hoped. The idea of integration was dropped, because the complexity of natural language, the complexity of its processing, and the complexity of natural language texts in general, were grossly underestimated (Smeaton, 1990). Although a great amount of research work had been put, for example, into the

development of sophisticated phrase analysis methods, it has been realized that simple keyword extraction techniques perform consistently better (Salton, 1986). As a result of this, researchers turned their attention to statistically-based methods for researching information retrieval processes.

Nevertheless, the latest results of research in this area—automatic translation between natural languages, natural language interfaces to databases, etc.—have indicated that this situation is changing; there is again a lot of work on re-applying techniques for *automatic natural language processing* to information retrieval problems. This research has concentrated mainly on semantic and syntactic construction of *term phrases*. It is known that the assignment of term phrases—rather than single terms—is helpful in providing narrow, specific identifiers when the original indexing vocabulary is too broad. An example of the linguistic approach is the study by Sparck Jones and Tait (1984) which has involved the use of a natural language parser to generate grammatically acceptable noun phrases from sentence-length natural language queries. These phrases can then be searched for in document abstracts.

However, there is still one problem to be solved with the phrase assignments: there is no easy way for generating only the useful identifiers and rejecting the useless ones (Salton, 1988). Since a reliable discrimination method is not currently available, a lot of work has still to be carried out in this area, as indicated by Fagan (1989). Recently, an interesting experiment has been described by Keen (1991a) who tested the use of term position information in non-Boolean retrieval systems. The best performance results were achieved by computing proximate matching term pairs in sentences plus a distance component. This supports the idea of the further development of term position devices for use in retrieval experiments.

As pointed out above, unsatisfactory results in the linguistic approach to automatic indexing in the 1960s have strengthened research interest in the use of *statistically-based techniques* for automatic term selection. A starting-point for this research is based on the hypothesis that the frequency of occurrence of distinct words in a natural language text is correlated with the importance of these words for the content repre-

sentation. Specifically, if all words were to occur randomly across the documents of a collection with equal frequencies, it would be impossible to distinguish between them using quantitative criteria.

One of the first to suggest that words occur in natural language text unevenly, and therefore, classes of words are distinguishable by their occurrence frequencies, was Luhn (1957):

> "A notion occurring at least twice in the same paragraph would be considered a major notion; a notion which occurs also in the immediately preceding or succeeding paragraph would be considered a major notion even though it appears only once in the paragraph under consideration. Notations for major notions as just defined would then be listed in some standard order as representative of that paragraph" (cf. Salton, 1986, p.1).

Using the constant rank-frequency law of Zipf, based on a general "principle of least effort" (i.e., author tends to repeat certain words instead of coining new and different words; the most frequent words tend to be short function words, e.g., AND, OF, BUT), Luhn suggested that a term selection procedure should be based upon the *collection frequency* of each keyword. Thus, he introduced the concept of the so-called "resolving power" of the index words extracted from document texts which should identify relevant items and distinguish them from non-relevant documents in a collection. Luhn (see Salton and McGill, 1983) emphasized that high-frequency terms are often non-specific and unable to discriminate sufficiently between relevant and non-relevant documents; very low-frequency terms can be good indicators of document relevance, but they contribute relatively little to the retrieval activity since they are most unlikely to be specified in a query. The concept of "resolving power" is therefore based on those index terms that are neither too rare not too common, i.e., having intermediate frequencies of occurrence.

Experiments, following these original ideas, have shown that they are not usable as stated in a practical operational retrieval environment. Salton and McGill (1983) pointed out the following limitations of the simple collection frequency approach:

- the elimination of high-frequency words may produce losses in recall;

- on the contrary, the elimination of low-frequency terms may produce losses in precision;

- there is no objective criterion for the selection of thresholds in order to distinguish the useful medium-frequency terms from the remainder.

In addition, as it is noted by Salton (1986), words occurring frequently in the texts of particular documents could not be used to distinguish these documents from the remaining texts of a collection if their occurrence frequency were also high in all other available documents.

The above insights were used to produce several weighting models for term selection, including a document frequency scheme, a signal-noise ratio model, and a term discrimination model.

The *document frequency* model is based on the calculation of how frequently a particular term occurs within both the text of an individual document and a document collection. The assumption is that a good term should have a high term frequency in a particular document, but a low overall frequency in the collection. Thus, good indexing terms are those whose occurrences are restricted to a relatively small number of documents (Salton and McGill, 1983).

The *signal-noise ratio* model (Salton and McGill, 1983) is based on data about the "concentration" of a term in the document collection. For a perfectly even distribution, when a term occurs an identical number of times in every document in a collection, the noise is maximized. Thus, a relationship exists between noise and term specificity, because broad, non-specific terms tend to have a more even distribution across the documents of a collection, and hence high noise. In contrast to specific terms, they do not contribute to a reduction of uncertainty about the document content. This model favours terms with very low document and collection frequencies, and in a retrieval environment usually distinguishes one or two specific documents from the remainder of the collection.

The third approach to the automatic selection of indexing terms is known as the

term discrimination model (Salton, 1975) which measures the degree to which the use of the term will help to distinguish the documents from each other. The model suggests that the ideal retrieval environment would be multi-dimensional index term space in which all of the documents are as far apart as possible. The discrimination abilities of different terms can then be evaluated by the change in the inter-document separations when a term is, and when a term is not, used for indexing. A good term will be one that helps to increase the separation of all of the documents, while the assignment of a poor term will tend to decrease inter-document separations, i.e., to increase space density. Results of experiments over several years (Salton and McGill, 1983) have shown surprisingly similar conclusions to those noted by Luhn (1957) that the best discriminators are medium frequency terms in the collection in which they occur. However, the results of the experiments carried out by Biru *et al.* (1989) indicated that medium frequency terms are not necessarily the best discriminators when relevance data is available; while these terms may include some of the best discriminators, they also include those with very poor discriminatory abilities.

Although some other models for term selection are reported in the literature (Salton and McGill, 1983), a lack of a general agreement about the most appropriate strategy is more than evident. While the term discrimination model suggests the use of indexing words with medium frequency of occurrence and the signal-noise ratio model proposes words with low document and collection frequencies, the document frequency model defines as the most useful words with high document frequency but low collection frequencies.

A failure to find the most appropriate means for automatic selection of indexing terms has consequently led to the current trend of using all of the keywords from a document or query text, and then refining them with an appropriate weighting scheme at search time (Willett, 1988a). The only exceptions are some of the very high frequency terms which are eliminated by means of a *stop-word list*. This list, also known as a negative dictionary, usually contains so-called non-content bearing words, i.e., function words (AND, OR, WITH, FOR, etc.), words from phrases that happen to be a part of a query (I WOULD LIKE, HAVE YOU GOT, etc.), and also specialty words in

the particular databases (e.g., LIBRARIES, in a database containing documents about librarianship). Although in the English language these stop-word lists usually contain about 300 common words, it is said by Salton and McGill (1983) that these terms comprise 40 to 50% of the text words; this is owing to the hyperbolic character of the Zipf law relationship. Thus, elimination of these terms increases the separation of all of the documents and also contributes to a reduction in the dictionary size of the inverted file.

It should be stressed that the automatic selection of indexing terms consists merely of *single* words. Starting from the notion that single term sets extracted from documents can offer only a simplified picture of the actual text content, suggestions for more refined content identifiers have also been put forward. On the one hand, in order to improve the recall performance of single terms (particularly those with low frequencies) the possibility of adding related terms derived from thesaurus and word association maps has been under investigation for a long time (Salton, 1986). On the other hand, in order to improve the precision performance of single terms (particularly those with high frequencies) the idea of identifying term phrases has also been seriously considered (Salton and McGill, 1983). Despite many experiments, the expectation of research in these two areas has not been met (Salton, 1988). It is known, for example, that the manual construction of a thesaurus is an enormous task and difficult to implement in operational environments in some subject areas. In addition, results of experiments in the automatic creation of a thesaurus have shown that only 20% of automatically derived associations between pairs of terms are semantically valid (Salton, 1986). Finally, the current experimental results on the automatic identification of term phrases are not sufficient to determine whether this approach does indeed result in increases in system performance; it remains to be seen how such processing can be best employed in information retrieval systems.

It follows that up to the present, single-term indexing methods have performed much better than other, sophisticated approaches to term extraction. In addition, the implementation of word conflation techniques has led to an increased recall performance for single terms.

1.2.3 Word conflation

One of the main problems encountered in automatic indexing and searching is the variation in semantically related word forms in free text. The differences are mainly caused by the requirements of grammar in a particular language, e.g., BIBLIOGRAPHY and BIBLIOGRAPHIC in English (or BIBLIOGRAFIJA and BIBLIOGRAFSKI in Slovene). The main problem to be solved is therefore to reduce the variants of a word to a single canonical form: this process is known as *conflation*.

Conflation is discussed in detail in Chapter 2.

1.3 Best-match searching

1.3.1 Comparison of conventional (Boolean) and best-match retrieval

Nearly all current document retrieval systems use *Boolean operators* (AND, OR, and NOT, possibly augmented by truncation and word proximity functions) in the searching procedure. Despite some advantages in linking chosen search terms by Boolean operators (e.g., the possibility of constructing highly specific and discriminating queries) there are some serious limitations to the conventional Boolean retrieval model. The main disadvantages, as discussed by Stibic (1980), Salton and McGill (1983), Cleverdon (1984), Willett (1985) can be summarized as follows:

- The formulation of the query, using the Boolean operators AND, OR, and NOT is usually a very difficult task for end-users with little experience. Simple features such as the conceptual difference between the AND and OR operators, as against the day-to-day usage of the words "and" and "or", and the fact that OR is also used in the inclusive meaning, i.e., it implies not only "either" but also "both", can cause many problems to the great majority of end-users. Thus, to access large databases or to search for complex topics the assistance of trained intermediaries is required.

- Searchers have only a limited degree of control over the size of the output that is produced in response to any particular query. Without a detailed knowledge of a particular database a broad query may involve the retrieval of many hundreds of documents while too detailed a query may lead to the retrieval of no documents at all. In both cases, a considerable amount of query reformulation is needed to obtain an appropriate volume of output. This is, of course, a very expensive and time-consuming process.

- A Boolean search results in a simple division of the particular database into two separate subsets: those records that satisfy the query and those that do not. Consequently, all items in the matching subset have an equal probability of being relevant to the searcher. Thus, in the search for relevance it is necessary to inspect the entire output list: if 100 records are retrieved, the last record seen by the searcher has an equal probability of being as relevant as the first. This means that conventional retrieval provides no mechanism for presenting output in decreasing order of probable relevance.

- In a Boolean search, there is no obvious means by which one can weight the terms in a query to reflect their relative degree of importance in the search. Boolean search—although many studies have shown that it is very easy and beneficial to calculate weights—assumes that all terms have weights of either 1 or 0, depending upon whether they happen to be present or absent in the query.

As a result of the limitations of conventional Boolean retrieval systems, there has been an increased interest in the use of *best-match searching techniques*. In this search procedure—also known as nearest neighbour or ranked-output search—the set of keyword stems resulting from the query-input module is matched against the sets of stems corresponding to each of the documents in the database (Willett, 1988a). When similarity between the query and each document is calculated—for example, the number of terms in common as proposed by Cleverdon (1984) in the so-called coordination level search—the retrieved documents are sorted into order of descending similarity with the query. The output from the search is a ranked list in which those documents which

the program judges to be most similar to the query are at the top of the list (i.e., "tip of the iceberg" as described by Stibic, 1980) and thus displayed first to the user. Since measures of similarities are usually based on formulae derived from probability theory, the documents at the top of the list are those with the greatest probability of being relevant to the query. Best-match retrieval is therefore also known as *probabilistic* retrieval (Porter and Galpin, 1988).

Using best-match search techniques many of the problems associated with Boolean retrieval can be eliminated (Willett, 1985; Willett, 1988a):

- There is no need for an expert to compose Boolean expressions for end-users. Best-match retrieval is very attractive to the end-users since they need input only an unstructured list of keywords.
- Since end-users obtain a ranked list of documents there are no problems associated with control over the size of the output. End-users can easily regulate the recall and precision searching performance. Even if there are hundreds of documents as candidates for display, a quick, precision-oriented search may involve the inspection of only the first 5 or 10 documents in the ranked list while greater recall may be obtained by going further down the list.
- It is very easy for the system to take weighting information into account to determine the degree of similarity between the query and each of the documents.
- In addition, those weights may also be based on end-users' judgments of relevance on the retrieved documents, and then feedback relevance information can be incorporated if a second search is required.

To put the advantages of best-match searching into practice two crucial components are required:

- an efficient *nearest neighbour searching algorithm* to permit the calculation of the query – document similarities;

- an effective means of *weighting the terms in the query* so as to reflect their relative importance in discriminating between relevant and non-relevant material in the database that is being searched.

1.3.2 Implementation of best-match searching

Comparing the advantages and disadvantages of both Boolean and best match retrieval, it is surprising that the former is one that predominates and the latter is still a rarity in current operational retrieval systems. According to Willett (1985), there are two main reasons why nearest neighbour searching techniques are not generally implemented in retrieval systems:

- since conventional, Boolean systems have been in use for many years, there is not a great deal of willingness by both users and systems providers to develop alternative information retrieval techniques;
- the first experiments in best-match searching were based on the incorrect assumption that the matching function required the comparison of a query with each of the documents in the file in turn. Thus, computational expense was perceived to be too high for practical implementation of best-match searching techniques.

Therefore, the main question put to researchers in advanced information retrieval has been how to produce a ranked output without the need to scan all of the documents in a particular database. One of the main results of the experimental work over many years has been a conclusion that the *inverted file* organization which forms the basis for most current on-line Boolean retrieval systems may also be used for the implementation of best-match searching.

As noted by Willett (1988a), two main groups of best-match searching algorithms have been developed for inverted file organization, using the characteristics of databases as a criterion:

- algorithms implemented on external databases and therefore constrained by the facilities of the on-line host retrieval system; nevertheless, there is an increasing

interest in developing so-called combined searching techniques (i.e., hybrid search) which, for example, enable ranking of the output from a Boolean search (see, for example, Salton *et al.*, 1983);

- algorithms implemented on internal, in-house databases and therefore much more flexible.

The latter group has been more extensively studied—a review is given by Perry and Willett (1983)—and is also briefly described below.

The principal problem that must be addressed by the best-match searching algorithm is the identification of terms in common between the query and a document, using the inverted file organization. So far, several algorithms have been developed and the most efficient seems to be that due to Noreault *et al.* (1977) which involves the *addition* of query lists. Experiments have shown that, despite the fact that large numbers of coefficients are evaluated, this algorithm provides an extremely efficient means of obtaining information about the number of keys in common between the query and each of the documents (Perry and Willett, 1983). The addition of the query lists may be achieved by taking the following steps:

1. query lists are processed in sequence: when a document number (identifier) is encountered for the first time in a query list, a counter is allocated to the document and set to one;

2. this counter is incremented by one each time that the document is encountered in subsequent query lists;

3. when all of the lists have been processed, each of the counters will contain the number of terms common to the query and to the appropriate document.

In the literature, the following four advantages of this search algorithm have been reported (Perry and Willett, 1983; Willett, 1988a):

- despite the fact that large numbers of coefficients are evaluated, this procedure is very fast in operation since the set of query terms is inspected only once in order

to determine which inverted file lists should be used;

- the calculation of all of the similarities involves disc access only for the query lists;
- having the number of terms in common between the query and each of the documents, it is easy to evaluate the corresponding similarity coefficient and rank the documents in order of decreasing similarity with the query;
- this procedure can be refined by the weighting of query terms, i.e., by incrementing the counters by the weight of each query term rather than by one. In this case, each counter will contain at the end of processing the sum of the weights for those terms that are common to the corresponding document and to the query.

The results of experiments in best-match searching have shown that retrieval effectiveness can be improved by incorporating procedures for the weighting of search terms.

1.4 Weighting of search terms

At the heart of nearest neighbour searching is the possibility of discrimination among the documents of a collection. A ranked output can then be achieved by using some quantitative measure of similarity between the query and each of the documents in the collection. As stated by Willett (1988a), the *similarity measure* consists of two main integral components:

- the *term weighting scheme* as a means of assigning weights to each of the index terms in a query or a document to demonstrate their relative importance;
- the *similarity coefficient* which uses these weights to calculate the overall degree of similarity between a query and each of the documents in the file.

Results of experimental work have shown that the term weighting scheme plays a more important role in the effectiveness of document retrieval systems than the choice of a

similarity coefficient.

There have been many different approaches to the calculation of the *similarity coefficient* (Salton and McGill, 1983). Particular attention has been given to the so-called component-by-component vector product, consisting of the sum of the products of corresponding term weights for two vectors. When some term is absent from the query or the document then this term will not make any contribution to the similarity coefficient, i.e., the numbers of matching properties for two vectors are reduced. Since it is customary to ensure that the similarity coefficient remains within certain bounds, say between 0 and 1, the so-called cosine coefficient (Salton and McGill, 1983) was found very easy to compute and also appeared to be very effective in the retrieval.

As has already been emphasized, *term weighting schemes* are of particular importance for retrieving documents in strictly ranked order. Term weighting schemes may be used either to weight query terms, or document terms, or both. The effectiveness of advanced information retrieval systems has been significantly improved by concentrating on developing methods for the weighting of *query terms*, with the documents being characterized by binary, i.e., present or absent, indexing terms. However, there is still an interest in the question of whether the terms in documents should also be weighted (Salton and McGill, 1983; Salton, 1986). It seems that some additional experimental studies will be needed to confirm the usefulness of such an approach in increasing retrieval effectiveness. Therefore, methods for weighting query terms will briefly be outlined in this section.

When tracing developments in research work on term weighting schemes, starting from the first empirical studies to the present firm theoretical basis, the following concepts are of particular importance:

- the concept of collection frequency, or inverse document frequency (Sparck Jones, 1972);
- the concept of a term relevance weighting system, based on probability theory (Robertson and Sparck Jones, 1976) which enables a relevance feedback search;

- the concept of the use of relevance weights when no relevance information is available, i.e., when the initial search is carried out (Croft and Harper, 1979);
- the concept which shows the close relationship between inverse document frequency weights and relevance weights (Robertson, 1986).

The concept of *inverse document frequency* (IDF), or *collection frequency* (Sparck Jones, 1972), derives from the hypothesis that matches on non-frequent terms are more valuable than ones on frequent terms. This idea is based on the model of term specificity, which states that very frequently occurring terms are responsible for noise in retrieval. Since the main problem in retrieval is to select a few relevant documents from many non-relevant ones, in this scheme the search terms are given a weight inversely proportional to their collection frequency. The matching value of a term is thus correlated with its specificity and the retrieval level of a document is determined by the sum of the values of its matching items. Experimental studies, using the IDF weighting scheme—which can be implemented by extremely simple means—have given results which are superior for best-match searching to those resulting from the use of unweighted query terms (e.g., coordination level searching where no distinction is made between frequent and non-frequent terms in common). Many subsequent tests have also confirmed that this scheme, although in principle based on a very simple approach, provides very effective results.

The IDF weighting scheme is based on information about the distribution of terms in documents in a particular collection. Thus, as stated by Willett (1988a) the IDF weight is characteristic of a particular term, and is not specific to a particular request. In a search for improving information retrieval the possibility of including relevance information in the weight assigned to a query term has also been investigated. As a result, a *term relevance* weighting scheme has been proposed (Robertson and Sparck Jones, 1976) which reflects the degree to which the query term can discriminate between relevant and non-relevant documents. It is suggested that terms occurring predominantly in relevant documents should be assigned greater weights than those terms occurring predominantly in non-relevant documents. The use of relevance information as a means

for the weighting of query terms is based on *probability theory*. Thus, the assumption is made that index terms occur independently in the relevant and non-relevant documents. In other words, if the probability of the i'th term occurring in a document, given that the document is relevant, is p_i, and the corresponding probability for a non-relevant document is q_i, then the weight for the term should be

$$log\frac{p_i(1-q_i)}{q_i(1-p_i)}$$

If it is known exactly which documents in the collection are relevant and which not, this weight can be calculated, using the following formula:

$$log\frac{r(N-n_i-R+r_i)}{(n_i-r_i)(R-r_i)}$$

where:
N = the number of documents in the collection
n_i = the number of documents indexed by i
i = given term
R = the number of relevant documents for some query
r_i = the number of relevant documents indexed by i

The theory that leads to this weighting scheme also results in the specification of a similarity coefficient which corresponds to the sum of the weights for those query terms which occur within a document. Thus, if d_i denotes the presence ($d_i = 1$) or absence ($d_i = 0$) of the i'th query term in a document, the matching function used is

$$\sum d_i log\frac{p_i(1-q_i)}{q_i(1-p_i)}$$

where the summation is over all of the terms in the query.

When full relevance information is available, i.e., where r and R are known for each term and query, these weights have been shown to give excellent retrieval performance. For example, the concept of the *relevance feedback* search, which is based on previously supplied user judgements on the relevance of displayed documents, is a very attractive tool for the new generation of free-text retrieval systems.

In the normal course of events relevance information is not available. Such a situation, usually during the *initial search*, when the user has not yet had a chance to

provide any relevance information to the system, has been considered by Croft and Harper (1979), also using the probabilistic retrieval model. They suggested that p_i and q_i (as defined by Robertson and Sparck Jones, 1976) should be estimated as follows:

- p_i should be assumed constant; the hypothesis is that all of the query terms have equal probabilities of occurring in relevant documents. Thus,

$$C = \frac{p_i}{(1 - p_i)}$$

  where C is a constant;

- q_i should be taken as the proportion of documents in the whole collection; the assumption is that the occurrence of a term in a non-relevant document may be approximated by its occurrence in the entire collection. Thus,

$$q_i = \frac{n_i}{N}$$

Under these circumstances, the weight of the term should be

$$log \frac{C(N - n_i)}{n_i}$$

Substituting into the relevance weight expression above gives

$$\sum d_i log \frac{C(N - n_i)}{n_i}$$

which may be expressed as

$$\sum d_i logC + \sum d_i log \frac{N - n_i}{n_i}$$

where the summation is again over all of the terms in the query. As can be seen from the above expression, this probabilistic model consists of two parts, and is therefore also known as the *combination match*. The first part corresponds to a simple coordination level match, i.e., the number of common terms between the query and the document, multiplied by the constant *log C*. Since it is reasonable to suppose that the significant words are those with potentially high probabilities of occurring in the relevant documents, i.e., almost equal to 1.0, Croft and Harper (1979) suggested on the basis of the

results of experiments that p_i should be 0.9 (giving a value of 9.0 for C). The second part of the expression is almost identical to the IDF weight, as described by Sparck Jones (1972).

Having the sum of the weights for those query terms which occur in each document, a rank list of documents in order of decreasing similarity to the query is displayed to the user, who can thereafter perform a relevance feedback search, giving relevance judgments for each displayed document. This second search should reflect user requirements more closely than the initial search.

Finally, an important contribution to the theoretical basis of term weighting schemes can be found in a recent article by Robertson (1986). Using again the probabilistic retrieval model, a comparison of the so-called point-5 relevance formula (i.e., a slightly modified formula of Robertson and Sparck Jones, 1976) and the formula of Croft and Harper (1979) demonstrated a very close relationship between IDF weights and relevance weights.

On the grounds of their successful performance, different term weighting schemes—mainly based on a probabilistic retrieval model—are implemented in current operational best-match retrieval systems which are therefore also known as probabilistic retrieval systems.

1.5 Conclusions

Besides automatic indexing, best-match retrieval and term weighting which form the nucleus of advanced document retrieval systems, there are also the following, very active areas of research by which development of these systems may be affected in the future:

- cluster analysis, or automatic classification (McCall and Willett, 1986; Griffiths *et al.*, 1986; Willett, 1988b);
- knowledge-based approach, i.e., automatic natural language processing and expert intermediary systems, as already described in this chapter;

- serial text searching, based on the use of the text signatures and parallel processing (Pogue and Willett, 1984; Carroll *et al.*, 1988; Willett, 1988a).

Research work in advanced information retrieval is now beginning to be reflected in operational systems of various sorts. These systems have already been shown to be able:

- to retrieve larger amounts relevant material than conventional systems;
- to replace trained intermediaries by end-users with limited experience of the search process.

One of these systems is INSTRUCT (INteractive System for Teaching Retrieval Using Computational Techniques) which was initially designed at the Department of Information Studies, University of Sheffield, for demonstrating advanced information techniques to students of librarianship and information science, but is becoming a useful basis for testing a range of research problems in information retrieval. The processing routines in INSTRUCT are, in very large part, independent of the actual language in which the texts have been written. The only exception is the stemming algorithm which has to take account of the morphological structure of the particular language. The original version of INSTRUCT is thus based on the stemming algorithm developed for the English language by Porter (1980).

Since the use of nearest neighbour searching techniques for providing end-user access to databases of Slovene text will be tested by the employment of the INSTRUCT package there was a need for an appropriate stop-word list and stemming algorithm. A review of stemming algorithms is given in the next chapter. On the basis of this review—along with the morphological characteristics of the Slovene language—the most suitable techniques for the development and design of the Slovene stemming algorithm will be selected.

Chapter 2

Automatic Word Conflation

2.1 Introduction

When designing both more effective and efficient retrieval systems it is necessary to develop techniques which will be able to cope with the morphological variations of natural language vocabulary. This is particularly important for reference retrieval systems, where documents are usually described by the words in the document title, keywords, and possibly by the words in the document abstract. A failure to account for the morphological complexity of terms can cause a substantial decrease in information retrieval performance.

The variations in words are mainly caused by the requirements of grammar in a particular language, e.g., MORPHOLOGY and MORPHOLOGICAL, from national usage, e.g., differences between American and British spelling (LABOR, LABOUR), and from mis-spellings. However, in many languages, including English and Slovene, terms with a common stem will usually have similar meanings, as for example:

Slovene	English
BIBLIOGRAFIJA	BIBLIOGRAPHY
BIBLIOGRAFIJE	BIBLIOGRAPHIES
BIBLIOGRAFSKI	BIBLIOGRAPHIC

Consequently, the performance of an information retrieval system can be improved

if these word variants are reduced to a single canonical form, without altering their meaning. This may be done by the removal of the various suffixes -Y, -IES, and -IC in English to leave the single stem BIBLIOGRAPH, and similarly by the removal of the various endings -IJA, -IJE, and -SKI in Slovene to leave the single stem BIBLIOGRAF. A procedure which uses systematic abbreviation of words so as to bring together words which are morphologically related, in the hope that they will also be semantically related (Walker and Jones, 1987), is known as a *conflation procedure*.

Word conflation performs two useful functions in information retrieval systems (Willett, 1988a):

- it may reduce the total number of different words, consequently leading to a reduction in dictionary size and updating problems;
- the retrieval effectiveness, particularly the recall performance, may be increased with the identification of semantically related terms.

Recent research in information retrieval has been more concerned with performance improvement than with storage reduction (Harman, 1987).

Conflation of word variants can be achieved in information retrieval either *manually* or *automatically*. An example of manual term conflation is the use of right-hand truncation at search time as specified by the searcher. Although considerable experience is needed if effective truncation is to be achieved, the majority of current on-line systems use this technique to reduce morphological differences between similar words. Since a certain amount of linguistic knowledge and training is required to perform right-hand truncation, this task must be carried out by an experienced intermediary and not by a casual end-user. Even when end-users are given the opportunity of employing truncation during searches, for example in experiments by Markey (1983), they consistently avoid the use of this facility.

Although manual word truncation is performed by an experienced intermediary, two major types of errors are still possible (Lovins, 1971):

- over-truncation occurs when too short a stem remains after truncation; this may

result in completely unrelated terms being conflated to the same stem, as with STRATEGY and STRATIFICATION being retrieved by the stem STRAT* (similarly in Slovene STRATEGIJA and STRATIFIKACIJA being retrieved by the stem STRAT*);

- *under-truncation happens when too short a suffix is removed and may result* in related words being described by different stems, as with COMPUTERS being truncated to COMPUTER*, rather than to COMPUT*, which would also include words such as COMPUTING, COMPUTATION (similarly in Slovene RAČUNALNIKI being truncated to RAČUNALNIK*, rather than to RAČUNAL*, which would also include words such as RAČUNALNIŠTVO).

Both types of error can significantly decrease retrieval performance, the former reducing precision of the search, the latter decreasing recall. For this reason, major bibliographic database vendors provide devices for intelligent review and selection of candidate terms from the dictionary file. Usually, an experienced intermediary performs a truncation and then adds (using Boolean OR) to the original search term(s) selected terms from the alphabetically sorted display. The result is a *term group*, consisting of the original search term and its variant forms.

An alternative way of bringing together semantically related word variants is the use of a conflation algorithm as part of the automatic computational procedure. The most common automatic conflation procedure is the use of a *stemming algorithm* which,

> "...reduces all words with the same root (or, if prefixes are left untouched, the same stem) to a common form, usually by stripping each word of its derivational and inflectional suffixes" (Lovins, 1968, p. 22).

In most cases prefixes are not removed because they tend to have a more drastic effect on the meaning of word than do suffixes; in addition, algorithmic techniques for the removal of prefixes are less well studied. The stemming algorithm performs, therefore, a similar function to that of manual right-hand truncation.

The obvious question, in terms of retrieval effectiveness, relates to a difference between the use of manual right-hand truncation and automatic word conflation. This

question was addressed by Frakes (1984), who found no significant difference between these two conflation procedures. Thus, he concluded, word conflation can be automated in retrieval systems with no average loss of performance, consequently allowing easier end-user access to the system. As a result of research work in automatic word conflation there are several systems which are based on stemming algorithms, for example CITE (Ulmschneider and Doszkocs, 1983), MASQUERADE (Brzozowski, 1983), MARS (Niedermair *et al.*, 1985), INSTRUCT (Hendry *et al.*, 1986a,b), and OKAPI (Walker and Jones, 1987).

Since the model of a Slovene language-based free-text retrieval system will be developed using INSTRUCT (Hendry *et al.*, 1986a,b), it is necessary to emphasize that the operation of INSTRUCT is language-independent, with one exception, i.e., the stemming algorithm. In this chapter, therefore, both a theoretical background of automatic word conflation and a methodological framework for the design of a stemming algorithm for Slovene will be outlined. This will provide a basis for the experimental work that is needed to develop a stemming algorithm for the Slovene language.

This chapter begins with a description of the main characteristics and types of conflation algorithms. This will be followed by a review of some of the conflation techniques which have been developed mainly for use in information retrieval. However, since word stemming has also found its application in the areas of natural language processing and computational linguistics, some of these techniques (e.g., morphological analysis of terms, as described by Cercone, 1978) will also be described as potentially interesting for information retrieval. In the third section some methods and results of the evaluation of conflation algorithms will be presented. The final section will serve as a starting point for the design of a stemming algorithm for the Slovene language.

2.2 Characteristics of stemming algorithms

Stemming algorithms which have been developed over the last two decades differ in many ways. The following classification is an attempt to capture the main features of automatic word conflation.

2.2.1 Types of stemming algorithms

According to Ulmschneider and Doszkocs (1983), stemming algorithms can be broadly divided into two classes: stemming purely by morphological analysis of terms, and stemming by the application of suffix dictionaries. Purely *morphological* techniques are characterized by the removal of suffixes from words according to their internal structure. The algorithm analyses the morphology of the word string, and, guided by rules of term morphology, determines candidate locations in the string marking the boundaries of a suffix. The optimal suffix candidate (often the longest) is then removed or replaced. For example, the dependency of any letter's appearance in a word upon the letter preceding or succeeding its position can be exploited to determine the boundaries of word units (Hafer and Weiss, 1974). Although morphological analysis (see also Cercone, 1978; Niedermair *et al.*, 1985) has the potential for detecting both prefixes and suffixes, it is rarely used, firstly because of the difficulty of deriving comprehensive and reliable rules (Ulmschneider and Doszkocs, 1983), and secondly because a substantial amount of processing is usually required (Lennon *et al.*, 1981).

The most common approach to stemming employs a *dictionary of suffixes* along with *rules* for its use. When a word is presented for stemming, the presence of these suffixes is searched for at the right-hand end of the word. If a suffix is found to be present, and the set of rules is satisfied, it is removed or replaced by another string, using either the principle of iteration or longest-match assignment. All algorithms in this group require the construction of a suffix dictionary and the formulation of a corpus of rules defining the morphological context of the suffixes. It is this group of stemming algorithms whose characteristics are described below.

2.2.2 Compilation of a suffix list

All algorithms based on a suffix dictionary can be broadly divided into two classes depending upon the manner of their development. The first class consists of a time-consuming *manual* analysis of vocabulary and language behaviour, in order to construct a suffix list and to formulate a corpus of rules. However, this effort is generally associ-

ated with conflation results of a high quality.

The second class of algorithms is characterized by the *automatic* generation of the suffix list from the bodies of text. All word endings occurring more often than some predetermined threshold are selected as suffixes. Apart from the minimum manual intervention required, this approach can be adapted to different collections (Tarry, 1978), if it is assumed that word conflation depends on the type of the source where suffixes were found.

Lennon *et al.* (1981) revealed that fully automated methods performed as well as procedures which involved a large degree of manual involvement in their development. They suggested that while the manual evaluation of lists of possible suffixes and rules gives results of a very high quality, the length of time taken often makes this method impractical. Their results confirmed that the consequent reduction in implementation costs in automatic generation of suffixes was not achieved at the expense of a decrease in conflation performance.

2.2.3 Mode of operation of stemming algorithms

Stemming algorithms can operate on the iterative principle, longest-match principle, or on a mixture of both.

Iteration is based on the fact that suffixes are attached to stems in a certain order, usually in the following form: stem – derivational suffixes – inflectional suffixes. An iterative stemming algorithm is simply a procedure which removes suffixes (sometimes single letters or strings rather than true suffixes) one at a time, starting at the end of a word and working toward its beginning. For example, Porter's iterative algorithm (Porter, 1980) processes the word GENERALIZATIONS in four iterations: in the first step GENERALIZATIONS is stripped to GENERALIZATION, then in the second step to GENERALIZE, in the third step to GENERAL, and finally to GENER.

The *longest-match* principle states that within any given class of endings, if more than one ending provides a match, the one which is the longest should be removed in one iteration. In the above example, the longest suffix -ALIZATIONS would be removed

in one step. This principle is implemented in algorithms by scanning the endings in order of decreasing length. If a match is not found on longer endings, shorter ones are scanned.

Consequently, on the one hand, longest-match algorithms are often easier to program but require a much larger dictionary since they must include all compound suffixes in each order in which they can appear. Iterative algorithms, on the other hand, while they permit the use of a much shorter list of suffixes, tend to be difficult to design since a great many endings must be examined in the preparation of a list.

These advantages and disadvantages of both the longest-match and the iterative algorithms often play a crucial role in the design of a conflation procedure for information retrieval. A typical example of an algorithm based on the longest-match principle is described by Lovins (1968), while Porter's algorithm (Porter, 1980) uses the iterative principle.

2.2.4 Conditional rules

A further important feature of a stemming algorithm is whether it is *context-free* or *context-sensitive*. The latter feature requires rules to be incorporated into an algorithm. These rules prevent the removal of a given suffix or a class of suffixes if a given condition is not satisfied. Conditional rules are usually of a quantitative nature and thus involve a minimum length condition, as for example, do not remove the suffix if the resultant stem would be less than four characters long. There may also be some qualitative contextual restrictions, which usually include word-specific rules, e.g., remove -ED unless the word is UNITED.

In a context-free algorithm, conversely, no quantitative or qualitative restrictions are placed on the removal of endings, and thus any ending which matches a word is accepted for stripping.

2.2.5 Recoding rules

Algorithms may also include some *recoding* rules which are applied after stemming has taken place. Recoding rules make changes at the end of the resultant stem to achieve the ultimate matching of varying stems. For example, the word FORGETTING might well be stemmed to FORGETT by removal of the suffix -ING. This will not match the word FORGET because of the repetition of the terminal consonant, and so a simple recoding rule is to remove one of any such doublings at the end of a stem. Other recoding rules may be used to achieve better conflation by rewriting some stems. For example, terminal -Y may be replaced by -I to conflate word forms ending in -Y with other related words: this rule would apply to the word LIBRARY to retrieve LIBRARIES when the -ES suffix is removed. Another example is the changing of -PT to -B to conflate word forms ending in -B with grammatically related words which change the -B to -PT: this rule would apply to the word ABSORPTION to retrieve ABSORB and ABSORBENT.

The construction of a comprehensive set of context-sensitive and recoding rules is one of the most difficult and time-consuming parts in the design process of stemming algorithm. Although there is no doubt that better conflation can be achieved by using these rules, it is important to emphasize that,

> "...there comes a stage in the development of a suffix stripping program where the addition of more rules to increase the performance in one area of the vocabulary causes an equal degradation of performance elsewhere" (Porter, 1980, p. 131).

Thus, there is always a danger of the algorithm becoming more complicated than it need to be. One of the important features of the design of a stemming algorithm is to obtain a balance between the number of rules, and simplicity and efficiency of processing.

As an alternative, or addition to the use of a set of predetermined recoding rules, some authors (for example, Dawson, 1974) suggest the use of a *partial matching* procedure at search time. This should allow a pair of non-identical stems to be matched if they are in large part similar. ABSORB and ABSORPTION can again serve as an

example, since they are similar in the first five letters. This approach is also related to *string similarity measures* which are described by Angell *et al.* (1983). These measures were mainly developed as part of experiments on automatic spelling corrections. Correction programs typically operate by searching in a large machine-readable dictionary for a word from the text to be corrected. If the word is not present, than it is assumed to be mis-spelt and some procedure is adopted that converts the mis-spelt word into a word that is present in the dictionary. The numbers of n-grams common to pairs of words (Freund and Willett, 1982), or the SPEEDCOP algorithm (Pollock and Zamora, 1984) are representatives of similarity-based techniques.

2.2.6 Users' needs

When developing a stemming algorithm it is also very important to consider its application area, i.e., whether it is going to be used in a specialized on-line reference retrieval system or in an on-line catalogue. This difference refers to the design of so-called "strong" stemming (i.e., longer classes of endings are removed from the word) or so-called "weak" stemming (i.e., only plurals and singulars are removed; this algorithm is also known as the "S" algorithm – Harman, 1987).

Since typical queries in specialized on-line reference retrieval systems usually include at least three words, and often more, it may not matter if some of the terms contain a substantial proportion of false drops attributable to stemming. Consequently, strong stemming can increase recall in these systems without unduly decreasing precision, as is shown, for example, in experiments by Frakes (1984). In addition, users of these systems are usually interested in an exhaustive search, and are prepared to examine even a substantial proportion of irrelevant material which could be caused by the use of "strong" stemming.

On the other hand, the use of a general on-line catalogue differs very much from the use of a reference retrieval system. As shown in experimental tests by Walker and Jones (1987) most search statements consist of only one or two words. Thus, even using only "weak" stemming can lead to unexpected results (e.g., compare the words RIGHT and

RIGHTS). In addition, many library users do not want an exhaustive search; they want to find one or two relevant items, and are not prepared to look at dozens of irrelevant records before they find them.

On the basis of the above assumption, the idea of a *multi-level* conflation system is advocated by Walker and Jones (1987), and also implemented in the OKAPI project, which is concerned with the design of an on-line public access catalogue at the Polytechnic of Central London. In the multi-level conflation procedure, the main difference is made between weak and strong stemming. While in the weak stemming, only plurals and singulars are conflated, the strong stemming removes longer classes of either derivational and inflectional suffixes. In addition, to avoid a good deal of noise in information retrieval, strong stems are given lower weights than weak stems, and thus, such records are displayed after the records retrieved on weak stems.

2.2.7 Language dependency of a stemming algorithm

An important characteristic of a stemming algorithm is its language dependency, which includes both the national usage of a language and the use of a professional terminology within individual languages.

Although there are some exceptions (see, for example, Jäppinen *et al.*, 1985), the great majority of stemming algorithms are designed for English language environments, i.e., they use suffix dictionaries composed of English word endings. However, the crucial part of these algorithms consists of different methods and techniques (i.e., modes of operation, the use of conditional and recoding rules) which can be applied to any other language where the semantic significance is contained in the stems, and not in the suffixes. Accordingly, the grammatical characteristics of the individual language, particularly the level of its morphological complexity, determines the adoption of these conflation techniques.

In this context, it is of a particular interest to note that the German language is characterized by a large number of compound words. Thus, as noted by Fuhr (1990), stemming techniques which were designed for the English language, and are

also dictionary-independent, could not produce successful results on German texts. Instead, a great amount of work has been devoted to the morphological segmentation of these compound words. For example, Wenzel (1980) has discussed the use of segmentation techniques analogous to those described by Hafer and Weiss (1974).

Within any national language it is also important to consider potential variations of stemming performance across different subject areas. While most algorithms are generalized in approach, some of them were also developed with a special emphasis to a particular subject area. Such an example is a stemming of medical English, as described by Ulmschneider and Doszkocs (1983).

2.2.8 Some other characteristics of stemming algorithms

In any suffix stripping program for information retrieval work, two points must be borne in mind. Firstly, the suffixes are being removed simply to improve information retrieval performance, and not as a linguistic exercise (Porter, 1980). It is not at all necessary that reduced morphological variants coincide with a linguistically correct root or stem, or that the character strings to be removed need be linguistically accepted affixes. The main problem to be solved using stemming is to reduce the variants of a word to a single canonical form, without altering the meaning.

The correctness of this approach was confirmed in experimental tests by Frakes (1984). His hypothesis was based on the linguistic theory that the terms truncated on the right at the root morpheme boundary will perform better than terms truncated on the right at other points. The hypothesis was not realized because small deviations from root boundaries did not significantly affect retrieval performance.

The second point to be considered in the design of a stemming algorithm is the fact that the success rate for suffix stripping is always less than 100% (Porter, 1980). In the development of a stemming algorithm, there is usually a temptation to deal with word forms which appear to be important but which are rare in most applications. An attempt to cope with such cases can result in the addition of many rules which can lead to a complexity in the program. In view of the error rate that must in any case be

expected, it is not worthwhile trying to cope with such cases. As already stated, the main aim of a stemming procedure is to obtain a balance between the number of rules and the simplicity and efficiency of processing.

Many of the characteristics of stemming described above are included in algorithms which have been developed over the last two decades, mainly for the English language environment. In the following section, a condensed review of some of these algorithms is given. Apart from studying the original reports, both the comparative paper by Lennon *et al.* (1981) and the survey by Walker and Jones (1987) form the basis of this review.

2.3 Conflation algorithms: a review

2.3.1 Lovins

One of the first conflation algorithms to be developed and tested was part of Project Intrex (Overhage and Reintjes, 1974) which was used for extensive experiments in on-line retrieval of library-type material. Word stemming was incorporated into this system to improve the effectiveness of information retrieval.

Lovins (1968), who participated in this project, obtained a preliminary list of endings by examining suffixes of a small portion of words in the Project Intrex catalogue and by studying a list of endings used at Harvard. The preliminary list was evaluated by applying the endings to dictionaries of normal and reversed English words to see whether the removal of a given ending would result in (1) two different stems matching, or (2) a stem not matching another stem which it should match. Either of these conditions necessitated the addition of new endings, the disposal of old ones, or the addition of context-sensitive and recoding rules.

This manual assessment resulted in the final list which contained about 260 endings, divided into 11 subsets; the subsets were ordered in accordance with the decreasing length of the endings and were internally alphabetized for easy handling. In this longest-match algorithm, each suffix was associated with one of 29 context-sensitive rules; there

were also 34 recoding rules to cope with words such as METER and METRIC.

Lovins' algorithm was also employed in some other experimental information retrieval systems, one of them being MASQUERADE (Brzozowski, 1983).

2.3.2 Dawson

Dawson's algorithm (Dawson, 1974) is based on that developed by Lovins (1968), and thus a longest-match method is used.

However, in his design of the algorithm, Dawson found the initial list of about 260 suffixes to be incomplete, lacking most plurals and other combinations of simple suffixes. His inclusion of additional endings resulted in a list of about 1,200 suffixes. To avoid the problems of storage and processing time which could be created by this large suffix list, Dawson used the principle of reversing the suffixes (and word specific suffix removal conditions) and indexing them by length and by final letter.

Unlike most of the algorithms, Dawson does not use recoding, describing this process as being extremely unreliable. Instead, his algorithm works basically on a partial matching principle, i.e., words are matched if their stems are "nearly" identical. Thus, the provision is made for the matching of, for example, ABSORB and ABSORPT. This is done by having a set of standard stem endings which can be considered equivalent (e.g., -RB and -RPT, or -MIT and -MISS). Dawson included fifty of these stem ending classes and the basic principle of his algorithm is as follows: if two stems match up to a certain number of characters and the remaining characters of each stem belong to the same stem ending class, then the two stems are conflated to the same form.

Both the extensive suffix list and removal conditions were drawn up manually using a Key-Letter-In-Context (KLIC) index; a similar approach to that used by Field (1975).

2.3.3 RADCOL

Lowe *et al.* (1973) tested two stemming algorithms as part of the RADCOL project, i.e., the information storage and retrieval system developed by Informatics for the Rome

Air Development Centre. The first algorithm involved two passes through a single list of 95 suffixes, but this was rejected in favour of a single pass, longest-match algorithm containing a much longer list of 570 endings.

To obtain this list, a multi-stage process was used. First, the characters of the most frequent words in the index (i.e., words occurring more than 10 times) were reversed, and the reversed words were sorted into alphabetical order. Adjacent words in the ordered list were then compared and whenever a match of n characters was found, strings containing 1, 2, ..., n characters were written out to tape. Thus, the list would contain characters strings such as -G, -NG, and -ING. These strings were sorted, cumulated, and the most frequent endings used as the starting point for a manual selection of the final suffix list. The final list was completed by examining the effect suffix adoption would have on the words in the collection and by comparison with the suffixes from the Lovins' algorithm.

Although the algorithm developed by Lowe *et al.* (1973), uses a long list of suffixes, the longest-match procedure is simple in application since there are only two context-sensitive and three recoding rules.

2.3.4 INSPEC

The conflation algorithm, designed at INSPEC (Field, 1975), was developed for statistical studies on the frequency and growth characteristics of free language indexing vocabularies. Both the word ending lists and the associated rules were drawn up manually, utilizing a KLIC (Key-Letter-In-Context) index. The KLIC index was produced from the single index words assigned to the documents in the test database, with each word being filed under each of its constituent characters. The KLIC index was arranged in the alphabetical order of the filing letter, and included a frequency count for each word type, and also for each distinct word ending. This listing was then scanned manually to give the lists of endings and context-sensitive rules to be used in the automatic stemming procedure described below.

The INSPEC algorithm is a mixture of longest-match and iterative suffix removal.

Minimum stem length, recoding rules, and three-stage conflation are the main features which were designed to improve its effectiveness. Of particular interest is the three-stage conflation procedure. The first stage (Algorithm 0), which is partly iterative in character, removes very common endings, such as plural forms, and eliminates stop-listed words. Words which are not stopped are then passed to the second part of the algorithm (Algorithm 1) which carries out most of the suffix removal. In this stage, a longest-match routine is used in which each suffix has an associated set of context-sensitive rules and a minimum permissible stem length. In the final stage (Algorithm 2), the algorithm makes adjustments to the stem, usually on the basis of stem length.

Field (1975) claims that the use of a three stage process leads to a significant increase in the efficiency of word conflation. The idea of a multi-level conflation system was adopted by Walker and Jones (1987) in their project OKAPI.

2.3.5 Automatic generation of suffix lists

Lennon *et al.* (1981), in the course of their evaluation of conflation algorithms, extended the method used in the RADCOL Project to achieve entirely automatic generation of suffix lists. A vocabulary of reversed words was used to produce a list of word endings occurring more often than some predetermined threshold. This list can then be used in a context-free, longest-match procedure, which is also known as the *frequency algorithm*. There is no doubt that, on the one hand, context-free algorithms with no conditional or recoding rules are much simpler to develop and may also be more efficient at run time since no character matching need be carried out to determine the context. However, on the other hand, the potential disadvantage of such a simple approach is, as stated by Lennon *et al.* (1981), that the inclusion of a string such as -INGNESS in a suffix set necessarily implies that all of the constituent substrings, e.g., -NGNESS and -GNESS, will also be included and thus the proportion of potentially useful suffixes in the set is much reduced.

A related method was used by Tarry (1978) who generated several sets of equifrequent character strings from the ends of words using an algorithm similar to that

described by Lynch (1977) and Cooper and Lynch (1979). The algorithm is based on the *variety generation* technique, i.e., on the selection of character strings of variable length occurring with approximately equal frequencies, and with low sequential dependence in a given body of text. A starting point in generating such "symbol sets" is the assumption that character strings representing suffixes would occur more frequently than other terminal character strings. In addition, it is also assumed that letter dependency within words decreases at the boundaries of word units such as suffixes. Thus, by generating symbol sets from the backs of the words, utilizing the above assumptions, it is possible to produce a workable set of suffixes which are not necessary correct linguistic endings.

Tarry's algorithm works on the longest-match principle, using suffix lists, generated by the method described above. The algorithm has no restrictions on suffix removal other than that the remaining stems should be of a minimum length of three characters. The algorithm is context-free, and no recoding is carried out, nor is partial matching employed at the conflation stage. Tarry justified this approach by the desirability of eliminating the large amount of manual preprocessing required, both in the construction of the suffix lists, and in the formulation of the suffix removal rules. The additional advantage of this procedure might be in automatically determining subject-specific or language-specific lists of suffixes.

Tarry's algorithm was compared with the INSPEC algorithm (Tarry, 1978), and the difference in performance between the two algorithms was found to be quite small.

2.3.6 Hafer and Weiss

One of the conflation algorithms which also does away with a substantial amount of manual preprocessing, for example the drawing up of the suffix lists, was developed by Hafer and Weiss (1974). This algorithm is based on segmenting lexical text into stems and affixes.

The stemming technique employed uses the concept of successor and predecessor varieties which are the numbers of distinct letters succeeding and preceding a given

character string in a text corpus. The motivation for using these quantities is the fact that within a word, the *i*th letter is dependent to some degree on the i - 1 letters that precede it. Within a natural word unit (i.e., a term or affix), this dependence is quite strong and increases with increased *i*. But if the *i*th letter begins a new word unit, the dependence is greatly reduced, (e.g., -M in the word ANTIMATTER is not at all dependent on its four predecessors). Within word units the successor variety is low and tends to decrease from left to right, while at boundaries the successor variety rises. By calculating the set of successor varieties for a test word and noting the peaks, the word units can be detected, and therefore both suffixes and prefixes can be removed from words.

Hafer and Weiss (1974) tested a total of 15 segmentation strategies, ranging from simply segmenting a word whenever a successor variety exceeded some predetermined limit, to using entropy methods, in which each successor letter was weighted by its probability of occurrence. The strategy which performed best in comparison with manual affix removal required either both the successor and predecessor varieties to exceed a threshold value, or the successor variety to be negative.

It is claimed by Hafer and Weiss (1974) that segmentation by this method achieves accuracy at least sufficient for the purposes of word conflation, and that the retrieval results obtained with various test collections are identical to those obtained with algorithms incorporating more manual processing. Their additional argument is that this method allows the text corpus to determine the segmentation points, making it more adaptable to changes in a collection, or to a new collection.

However, in the evaluation experiments by Lennon *et al.* (1981), the Hafer-Weiss method performed worse than other tested algorithms. It was also found that this algorithm required a substantial amount of processing to determine the predecessor and successor varieties since the entire dictionary, and its reversed form, must be inspected for segmentation to take place.

2.3.7 SMART

The SMART system (Salton, 1971) uses an enhanced version of the Lovins stemmer that removes many different suffixes. The stemming algorithm implemented in the SMART system operates in the following way. First, the longest possible suffix is found that allows the remaining stem to be of length 2 or greater. The resulting word stem is then checked against an exception list for the given suffix, and, if passed, is processed into the final stem in a cleanup step. This recoding step uses a set of rules to produce the proper word ending, such as removing a double consonant. The algorithm uses auxiliary files containing a list of over 260 possible suffixes, a large exception list, and the recoding rules.

2.3.8 MORPHS

The retrieval system MORPHS, i.e., Minicomputer Operated Retrieval (Partially Heuristic) System, which incorporates automatic stemming for compact storage, document retrieval and automatic role indicating, is described by Bell and Jones (1976). Automatic stemming is based both on the standardization of word forms (mainly, plural forms are substituted by the singular) and on the use of so-called role indicators.

The automatic *role indication* relies on the fact that affixes of a word can provide information about the function of that word. In MORPHS, when an affix is stripped from a word, it is replaced by the role indicator associated with that affix. This forms a potential for searching either roots or derived forms. Thus, it is possible to search MIX; or MIX (role *A*) – implying MIXING; or MIX (role *D*) – implying MIXED.

Automatic stemming in MORPHS is based on the use of an extensive suffix list. The length of this list is due in part to the number of exceptions incorporated (thus, CATION and STATION are protected from the -ION stripping routine), and in part to the presence of chemical suffixes. The system attempts to guard against the removal of apparent suffix strings by a minimum stem length and by checking that the stem is present in the stem dictionary before the affix is removed. Both longest-match and iterative methods are used in the removal of suffixes and the creation of stem

dictionaries. For example, the word PREVULCANIZATION is stripped first to PREVULCANIS (*A*), then to PREVULCAN (*A*) and finally to VULCAN (*DA*); the letters within parentheses indicate the role indicator.

2.3.9 Cercone

Some conflation algorithms have been developed and tested for natural language applications. Since the design of such algorithms is primarily concerned with meanings and functions of words, there is a need for a higher degree of linguistic correctness, which usually results in a complicated stemming process. One of these algorithms, known as the *morphological algorithm*, was developed by Cercone (1978).

Cercone's algorithm aims to determine the root of a term by removing suffixes and prefixes, using the principle of iteration and consulting an affix dictionary. This process uses a system of order classes, assuming that affixes, and particularly suffixes, are attached to stems in a certain order. The removal of some affixes is followed by recoding of the root. After recoding, the root dictionary is searched and, if a match is made, the root and the affix are output. If there is no match, the next order class of affixes is accessed until a root is eventually found.

Thus, the algorithm requires a root dictionary containing all possible root forms, various affix lists, and the construction of recoding rules. All of these were drawn up manually. It is claimed by Cercone (1978) that such a morphological analyzer can significantly aid the identification of the function and meanings of words in a given corpus of text.

2.3.10 MARS

Niedermair *et al.* (1985) have developed a system called MARS which aims to facilitate the user's access to all searchable terms in a database which are morphologically related to the given search term. Linguistic knowledge and word decomposition procedures are at the heart of this system.

The operation of MARS, after the automatic elimination of stop-words, is based on the use of a morpheme dictionary and morpheme grammar which are employed to achieve the *morphological decomposition* of words, i.e., to split words into prefix, stem, derivational and inflectional elements. The extracted word stems are collected in a stem-file in which pointers back to the text words containing the particular stem can be followed, enabling retrieval of these words.

The morpheme dictionary contains affixes, inflectional endings, and fillers. They are all represented in a uniform way, that is, the string itself and a 32 bit-string indicating special morpheme characteristics and certain compositional properties. The morphemes in the dictionary are the longest possible strings obtainable from all of its possible derivations (TRADITIONALLY, for instance, would be viewed as a derivation of TRADITION and not of TRAD(E)). Two smaller lists are added to this dictionary; one includes "irregular" stems like Latin and Greek plurals and irregular verb forms, the other list contains strings which regularly undergo grammatical change, such as -Y to -IE.

Before the actual decomposition starts, a pre-processor checks to see if string transformations are necessary. After this, the three lists mentioned above are used by a decomposition grammar which deals with each word. After having reached a certain state in the word, say prefix, certain conditions have to be fulfilled if the word is to be passed to the next stage. These rules are coded in a morpheme grammar.

The retrieval performance of MARS was tested on a sample of twelve real searches. It is reported by Niedermair *et al.* (1985) that recall was increased by 68% compared to the total number of documents retrieved without MARS; this was achieved without a significant decrease in precision which dropped only by 7%. Retrieval tests also revealed that MARS performed less effectively when searching compound words, phrases and verbs.

2.3.11 Porter

Porter's iterative stemming algorithm was developed at the University of Cambridge Computer Laboratory (Porter, 1980). The following starting points guided a construction of this algorithm: the algorithm was developed to improve information retrieval performance; its design was not thought to be a linguistic exercise; a certain error rate was expected. The algorithm uses an explicit list of endings, and, with each suffix, the criterion under which it may be removed.

The most important part of Porter's algorithm is the concept of the "measure" of a word, which guards the removal of suffixes when the stem is too short. This measure describes the length and number of consonant–vowel–consonant strings present, a concept first studied by Dolby and Resnikoff (1964) to establish certain regularities in the structure of written English words. Since the employment of this concept greatly contributes to the simplicity and efficiency of Porter's algorithm it is described below in detail.

According to Dolby and Resnikoff (1964), a word is defined as a lexed item represented by a sequence of letters of the alphabet. Porter (1980) adopted their idea of two main sets of letters, i.e., a set of vowels (A, E, I, O, U, and Y preceded by a consonant), and a set of consonants. A consonant can be indicated by c, a vowel by v. A list $ccc\ldots$ of length greater than 0 can be denoted by C, and a list $vvv\ldots$ of length greater than 0 can be denoted by V. Any word, or part of a word, therefore has one of the four forms:

$$CVCV\ldots\mathrm{C}$$
$$CVCV\ldots\mathrm{V}$$
$$VCVC\ldots\mathrm{C}$$
$$VCVC\ldots\mathrm{V}$$

These may all be represented by the single form

$$[C]\ VCVC\ldots[V]$$

where the square brackets denote arbitrary presence of their contents. Using $(VC)^m$ to denote VC repeated m times, this may again be written as

$$[C]\ (VC)^m\ [V]$$

where m is the measure of any word or word part, i.e., *VC*. The case $m = 0$ covers the null word. Here are some examples, taken from Porter (1980):

$m = 0$	TR, EE, TREE, Y, BY.
$m = 1$	TROUBLE, OATS, TREES, IVY.
$m = 2$	TROUBLES, PRIVATE, OATEN, ORRERY.

The measure, m, is therefore used to help decide whether or not it is wise to remove a suffix. For example, -ATE is removed from DERIVATE (m is greater than 1), but not from RELATE.

Porter's algorithm is a five-step, iterative procedure, using a dictionary of about 60 suffixes. Step 1 deals with plurals and past participles, the subsequent steps are much more straightforward. The algorithm has only a few context-sensitive and recoding rules, and so is economical in computing time and in storage. Despite its simplicity, retrieval tests (Porter, 1980) showed that Porter's algorithm performed slightly better than the much more complicated procedure described by Dawson (1974).

The advantages of Porter's algorithm include its simplicity and efficiency of processing, its detailed description for easy implementation in any high-level programming language and its good performance in retrieval tests by Lennon *et al.* (1981). It has thus been implemented in several experimental retrieval systems, these including CATALOG (Frakes, 1984), INSTRUCT (Hendry *et al.*, 1986a,b), and OKAPI (Walker and Jones, 1987). The way in which Porter's algorithm was incorporated in the OKAPI system (Walker and Jones, 1987) is of particular interest and it is therefore described below.

2.3.12 OKAPI

OKAPI (Walker and Jones, 1987) is a system which was developed on the basis of results of on-line catalogue research at the Polytechnic of Central London. While the first version of OKAPI (OKAPI'84) was mainly concerned with the design of modules which could allow end-users to perform on-line searches, more recent work (OKAPI'86)

has concentrated on the improvement of subject retrieval. Consequently, the following three devices were incorporated into OKAPI: automatic stemming, automatic cross-referencing, and semi-automatic spelling correction.

OKAPI is reported to be the first on-line catalogue, accessing a general collection, the performance of which is based on automatic stemming. Since the use of uninhibited stemming in on-line catalogues can lead to a good deal of noise, as stated by Walker and Jones (1987), the idea of a *multi-level* conflation system was adopted in OKAPI. The actual stemming procedure used was that of Porter (1980), splitting it into two levels, i.e., weak and strong stemming.

In Stage 1 (weak stemming), regular English plurals and -ED and -ING endings are first removed, and then most double consonant endings reduced to single. In addition, no endings are removed from words under four letters long or from "words" which contain digits or other non-alphabetic characters. Specifically, Step 1 of the original Porter's algorithm is done, followed by a spelling standardization, which is mainly an attempt to cope with the differences between British and American spelling. In Stage 2 (strong stemming), endings given in Steps 2 to 5 of Porter's algorithm are removed. However, to avoid noise in performance, strong stems are given lower weights than weak stems, and those records are displayed after records retrieved with weak stems. An additional interesting feature of this procedure is that stems are never actually displayed to the user, since words often look strange when they have been stemmed.

Results of evaluation tests on the OKAPI system (Walker and Jones, 1987), based on a set of 255 searches, revealed that weak stemming is entirely beneficial in subject retrieval in on-line public access catalogues. A significantly higher proportion of relevant documents was retrieved than without stemming. On the other hand, strong stemming was not always found to be useful, although it behaved well more often than it behaved badly. Walker and Jones (1987) therefore suggested that strong stemming should not be used alone in a general catalogue; and when used with weak stemming, strong stems must be given lower weights than corresponding weak stems.

Since OKAPI is at present a unique representative of an on-line catalogue which uses

automatic stemming when accessing a general collection, the results of the experimental work by Walker and Jones (1987) can be very useful in improving access to any academic or public library database, as well as to some specialized bibliographic databases in which the content of documents is represented only by title and descriptors, and not by abstracts.

2.3.13 CITE

Another example of an on-line catalogue which uses a stemming procedure, but which is designed specifically for medical terminology, is CITE (Ulmschneider and Doszkocs, 1983). This catalogue enables access to the monograph collection at the National Library of Medicine in Betheseda, Maryland, where it is used by medical researchers and students. Its main features are as follows: it accepts queries in ordinary medical language, uses automatic stemming and synonym generation via MeSH (i.e., Medical Subject Headings), assigns weights to stems and headings which determine their relative importance, outputs records in ranked order, and also allows relevance feedback.

The stemming procedure in CITE shares many of the characteristics of the approaches to term conflation described above. For example, it employs a suffix dictionary along with application rules and is intended for English language text. However, a special additional emphasis is placed on the stemming of medical English and on its dependency on the MeSH structure.

A suffix dictionary was designed on the basis of an analysis of the terminal character strings of all unique terms in MEDLINE. To obtain this dictionary all MEDLINE terms were sorted by their terminating characters, and then compared to find matching strings. Each unique terminal string was then listed in reverse order along with its frequency of occurrence. The relative frequency of the terminal string itself, as well as the frequency of strings either containing the candidate string or contained in the candidate string, determined the construction of the final suffix list. In addition, many exceptional cases were also included in the suffix dictionary.

The stemming algorithm, which is iterative in operation, consists of the identifica-

tion of the word stem and the automatic selection of "well-formed" morphological word variants from the actual inverted file entries. These term groups are further enriched by controlled vocabulary indexing terms from NLM's Medical Subject Headings (MeSH) which also includes forms of synonyms.

Although CITE provides many advanced features in word conflation, there is no published data on its actual use or effectiveness (Walker and Jones, 1987). In general, as described below, there is an evident need for more comprehensive, particularly comparative and quantitative, evaluations of stemming algorithms for information retrieval to be carried out in the future.

2.4 Evaluation of conflation algorithms for information retrieval

The retrieval performance of stemming algorithms can be evaluated using test results of different levels of comparison. The most common levels are the following: a comparison between full word retrieval and retrieval using automatic stemming, a comparison between right-hand truncation and automatic word conflation, and a comparison between different stemming algorithms. The next three sections form a survey of some of the evaluation studies which have been published.

2.4.1 Automatic stemming vs. full word retrieval

Some of the retrieval tests (e.g., Lennon *et al.*, 1981; Niedermair *et al.*, 1985; Walker and Jones, 1987) have shown that conflation algorithms perform significantly better in information retrieval than the use of unstemmed words.

However, there is evidence, reported by Harman (1987), that the use of stemming algorithms does not necessarily result in improvements in information retrieval. Using three general purpose stemming algorithms (Porter, Lovins, and the "S" algorithm) on three different collections, her tests revealed no substantial difference between full word retrieval and retrieval using suffixing. Although individual queries were affected

by stemming, the number of queries with improved performance tended to equal the number with poorer performance, thereby resulting in little overall change for the entire test collection. Additionally, stemming caused a significant increase in query processing time. Despite these results, she concluded that

> "...the stemming of query terms is intuitive to many users, is more convenient than specifically using truncation and wildcard characters in queries, and is often necessary for helping queries retrieve relevant documents in the top ten or thirty documents" (p. 106).

2.4.2 Automatic stemming vs. right-hand truncation

Since term conflation is normally achieved in conventional on-line systems using right-hand truncation as specified by the searcher, the obvious question is whether word conflation can be automated in retrieval systems with no average loss of performance. The most comprehensive experimental study so far, addressing this question, was carried out by Frakes (1984) who found no significant difference between these two conflation procedures. He concluded that term conflation can be automated in a retrieval system, thus allowing easier end-user access to the system.

2.4.3 Evaluation of different conflation algorithms

Since automatic stemming can potentially increase retrieval effectiveness, the evaluation of stemming algorithms must consider the following points: the efficiency of operation (i.e., the number of conditional and recoding rules, the amount of processing time required); the ease of implementation (i.e., a detailed description of an algorithm) and the amount of manual involvement in the development of an algorithm. Some of these points, as described in the above sections, were analyzed by Porter (1980) and Lennon *et al.* (1981).

However, the most important part of the evaluation of stemming algorithms relates to their main functions, i.e., to their *effectiveness* in:

- decreasing the size of dictionaries, and

- increasing retrieval performance.

The comparison of different stemming algorithms, considering both aspects of effectiveness, has so far been carried out only in retrieval tests by Lennon *et al.* (1981). In addition, the use of different strengths of stemming has recently been studied by Keen (1991b). There have also been some other studies reported, but they have been mainly interested in the comparison of only two algorithms, for example Porter's comparison of his and Dawson's algorithm (Porter, 1980). The results of retrieval tests by Lennon *et al.* (1981) are briefly described below.

Lennon *et al.* (1981) tested six conflation algorithms (i.e., INSPEC, Lovins, RADCOL, Porter, Hafer-Weiss, and the so-called "frequency algorithm") on several databases. Their starting point for evaluation was a notion that errors in stemming algorithms can be of two types: either a word can be understemmed, in which case too little of the word is removed, or it can be overstemmed, when the converse applies. Consequently, understemming leads, on the one hand, to the omission of relevant material, and therefore to lower recall, while overstemming, on the other hand, causes the retrieval of irrelevant documents, and therefore leads to lower precision. The amount of dictionary compression can be used as an indicator of both understemming or overstemming.

Compression results achieved in tests by Lennon *et al.* (1981) confirmed the existing correlation between overstemming and dictionary compression. For example, the RADCOL algorithm achieved the greatest compression (49.1%) but it tended to overstem, while Porter's algorithm achieved the least compression, but tended to understem.

Consequently, since stemming is mainly a recall-oriented device, it was expected that strong algorithms would tend to increase retrieval effectiveness more than weak ones, especially in a recall oriented search. However, retrieval tests demonstrated that there is no relationship between the strength of an algorithm and the consequent retrieval effectiveness arising from its use. For example, Porter's algorithm tended to understem, but it performed better than the RADCOL algorithm which tended to overstem. The INSPEC algorithm, on the other hand, is also a strong algorithm, but it gave the best precision-oriented search. Experiments by Keen (1991b) using different

strengths of stemming did not confirm average improvements in performance of any magnitude either.

2.5 Conclusions

Many nearest neighbour document retrieval systems have been described in the research literature and operational implementations of some of these ideas are now available (Willett, 1988a). To date, the great bulk of this work has been carried out with English language material, where stop-word lists and stemming routines have been available for many years. In order to be able to provide end-user access to databases of Slovene text—using the nearest neighbour searching techniques—there is a need for an appropriate stop-word list and stemming algorithm.

The only work which has been carried out in this area to date is the M.Sc. thesis of Dimec (1988), which reports a computer analysis of the frequency and growth characteristics of the Slovene language as a basis for the development of an automatic indexing system for Slovene medical literature. Two stop-word lists were created in this study; in addition, a simple stemming algorithm was designed, based on the use of a list of 381 suffixes. However, Dimec (1988) notes that there are many limitations with the procedure as described, these resulting in large part from the small number of suffixes used and from the very complex morphology of the Slovene language.

Thus, a Slovene language free-text retrieval system, based on the use of nearest neighbour searching techniques, demands the following:

- the creation of a general purpose stop-word list, which is not restricted to medical literature;
- the design of a more powerful stemming algorithm that takes greater account of the language's morphological structure.

In order to be able to accomplish these two objectives a detailed knowledge of the main characteristics of the Slovene language is required.

Chapter 3

Main Characteristics of the Slovene Language

3.1 Introduction

Slovene is spoken by some two million people living mainly in the western part of Yugoslavia, i.e., in the Republic of Slovenia, one of the six federal units comprising that country. In Italy (Trieste, Gorizia, Julian Venetia and Retia) and in Austria (Carinthia) it is spoken by Slovene minorities, elsewhere, especially in America, by Slovene emigrants.

Linguistically, Slovene is a South Slavic language with a speech area to the west of Serbocroatian, wedging into the Italian, German, and Hungarian linguistic territories in the extreme eastern spurs of the Alps. One of the main characteristics of *contemporary Slovene* is a profound discrepancy between its written and spoken form. Although the written Slovene language was created in the 16th century and intended for religious use at first, it slowly expanded its communicative function, via a wide range of dialects which differed from district to district. Indeed, it has retained an unusual degree of vigour and distinctiveness up to the present day. Therefore, contemporary Slovene literary language, prescribed for educated speech, described in its grammar books and

used in literature, scholarship and in the communication media, represents only its standard form and has been labeled as a "Schriftsprache", a "book language"; in other words, an artificial language (Lencek, 1982). In the literature, it is usually stressed that there is no such thing as standard spoken Slovene; that the gap between the natural language or dialects, and the written language is almost unbridgeable. This gap consists of such features as stress placement and tense/lax vowel quality, which are not reflected in the writing system (Bidwell, 1969); thus, the written form gives an impression of unity and consistency not actually present in the natural language. It is in this sense that the contemporary Slovene language is indeed more a book-language than any other Slavic language (Lencek, 1982).

However, literary Slovene represents the only common standard which unites the speakers of Slovene. This point is very important for a wide range of different socioeconomic, political, educational, and cultural activities in Slovenia, one of which is the design and development of information retrieval systems. Deriving from a need to design automatic word conflation procedures for the Slovene language it is, therefore, the main aim of this chapter to give a concise description of the main characteristics of contemporary Slovene language, particularly its inflectional morphology. On this basis, the essential requirements for a design of a stemming algorithm for Slovene will be defined.

This chapter consist of two main sections. In the first section, the Slovene alphabet and pronunciation are concisely outlined. The second section is much larger, and contains an analysis of the morphological structure of the Slovene language. First, the concept of word-formation is briefly summarized, followed by a detailed description of the Slovene inflectional morphology. Particular emphasis is given to the explanation of basic grammatical categories, word (formal) classes and morphemic alternations, occurring in both stems and suffixes during inflections. In the conclusion, the main points to be considered in a design of the Slovene stemming algorithm are outlined. Whenever possible, a comparison between Slovene and English grammar is also made, intended mainly to make the complexity of the Slovene morphology clearer to the English reader.

The literature which was consulted during the work on this chapter is listed in the Appendix A. However, it is necessary to emphasize that three sources in particular were most helpful, i.e., Lencek (1982), and Toporišič (1975; 1984). In addition, many examples which serve in this chapter to illustrate the structure of the Slovene language were taken from Toporišič (1975).

It is perhaps pointless to say that the sources listed in the Appendix A represent a far more exhaustive and comprehensive analysis of the Slovene language than the description presented in this chapter. Therefore, readers who are deeply interested in the structure of Slovene are strongly advised to use the literature included in the Appendix A.

3.2 The Slovene alphabet and pronunciation

Contemporary Slovene has twenty-five letters. Their order is as follows: a, b, c, č, d, e, f, g, h, i, j, k, l, m, n, o, p, r, s, š, t, u, v, z, ž. In foreign words the letters *q, w, x, y* may also appear; in the alphabet, *q* stands between *p* and *r*, *w, x, y* between *v* and *z*. As in the English alphabet, all letters can be divided into two main groups, i.e., vowels and consonants.

3.2.1 Vowels

The Slovene alphabet has five vowels: a, e, i, o, u. The pronunciation of the vowels in Slovene is quite complicated since there are three accent marks to be observed (i.e., ´ indicates long close vowels, ` indicates short open vowels, ^ indicates long open vowels), yet they are not placed over vowels in the written language. For example, *e* can be pronounced in the following ways:

- *é* (*verjéti - to believe*); it is a sound similar to the first part of the English word *aim*;

- *è* (*vsè* - *everything*); it is a sound similar to the first part of the English word *everything*;

- *ê* (*Vêra* - *Vera*, name); it is a sound similar to the middle part of the English word *man*.

In addition, *e* can also be pronounced as the unstressed *e* (for example, *dedek* - *grandfather*, where the second *e* is reduced, like the sound at the beginning of the English word *about*).

Because of the discrepancy between vowels in the written form and their vocalic sounds, it is not surprising that a common view expressed by foreign speakers is that Slovene pronunciation is "impossible" to learn (Tollefson, 1981).

3.2.2 Consonants

There are 20 consonants in the Slovene alphabet: b, c, č, d, f, g, h, j, k, l, m, n, p, r, s, š, t, u, v, z, ž. The Slovene consonants are mostly pronounced as they are spelled. The consonants which are different from English either in spelling or pronunciation are given below:

- *c* is pronounced as *tz*, as in the English word *tzar*;

- *č* is pronounced as *ch*, as in the English word *church*;

- *g* is pronounced as *g*, as in the English word *gun*;

- *h* is pronounced as the German *ch* (*Dach*) and not as the English *h* (*he*);

- *j* is pronounced as *y*, as in the English word *yet*;

- *l* preceding a vowel is pronounced as a middle or European *l*: *šola* – German *Schule*. The pronunciation of the final *l* is similar to the pronunciation of the English *w*;

- *s* is pronounced as *s*, as in the English word *sit*;

- *š* is pronounced as *sh*, as in the English word *show*;
- *v* has three sounds:
  - it is a true *v* (English *v*) when preceding a vowel;
  - it is pronounced as a *w*, before a consonant or if it is the final letter of a word;
  - sometimes it is a true *u* (English *u*), especially when both preceded and forwarded by consonants;
- *z* is pronounced as *z*, as in the English word *zero*;
- *ž* is pronounced as *s*, as in the English word *measure*.

For the purpose of a description of morphemic alternations caused by the Slovene inflectional morphology, it is perhaps useful to define a further functional classification of consonants. They can be divided into the following groups:

- SONORANTS: *v, m, n, r, l, j*
- OBSTRUENTS:
  - voiced: *b, d, g, z, ž*
  - voiceless: *p, t, k, s, š, c, č, f, h*

As will be outlined in the following sections, the position of sonorants or obstruents at the end of the stem plays an important role in the changes of both stems and suffixes during the inflection of words.

3.3 Morphological structure of the Slovene language

3.3.1 The concept of word formation

Before saying anything about the morphological structure of the Slovene language, it is necessary to introduce briefly—in order to understand the main prerequisites for a

design of the automatic word conflation—the concept of *word formation* in Slovene. Using a highly simplified notion (for a comprehensive analysis of the theory of word formation in the Slovene language see Vidovič-Muha, 1988), it can be stressed that formation of words in Slovene does not differ much from those languages where new word forms are created using a *stem* with the addition of *derivational suffixes*. Although many distinctive words can be created from one stem in this way, they still usually have a similar meaning, as for example:

Slovene	English
RAZISKAVA	RESEARCH
RAZISKOVALEC	RESEARCHER
RAZISKOVATI	RESEARCH

This feature of the Slovene language is extremely important because it indicates the use of *right-hand truncation* as the best way to achieve word conflation.

However, Slovene is characterized by a wider range of derivational suffixes than is English (for a comprehensive list of derivational endings see Toporišič, 1984). It is Slovene *inflectional morphology* which formally distinguishes both languages. This part of the structure of the Slovene language will be described in the following sections. Such an approach will be justified at the end of this chapter where all possible variants of the above stem RAZISKOVA- will be listed.

3.3.2 Inflectional morphology of Slovene

In order to be able to describe the morphological complexity of the Slovene language, it is necessary to introduce three main concepts: word (formal) classes, inflection, and grammatical categories.

The concept of basic *word classes* is at the heart of the Slovene morphology. According to Toporišič (1984), the Slovene language has been characterized by the following nine word classes:

1. substantive words:

   (a) noun (*hiša, otrok, tla*; *house, child, floor*)

(b) verbal noun (*dejanje, skrb, petje*; *action, care, singing*)

(c) substantival adjective word (*dežurna*; *on duty*)

(d) substantive pronoun (*jaz, ti, on, kdo*; *I, you, he, who*)

2. adjective words:

(a) adjective (*lep -a -o*; *pretty*)

(b) numeral (*en -a -o, prvi -a -o*; *one, first*)

(c) adjective pronoun (*tak -a -o, moj -a -e*; *such, my*)

3. verb:

(a) personal forms (*dela -m -te*; *I'm working, you're working*)

(b) descriptive participle ending in *-l*, and *-n/-t* (*delal, delan, ubit*; *worked, murdered*);

(c) non-inflectional forms (*delaje, delajoč, delati, delat*; *to work*)

4. adverb (*doma, včeraj, zakaj*; *at home, yesterday, why*)

5. predicate (*všeč, treba, tiho*; *wished for, it is necessary, quiet*)

6. preposition (*do, za, brez*; *to, for, without*)

7. conjunction (*in, toda, če*; *and, but, if*)

8. copula (*samo, tudi, pač*; *only, also, indeed*)

9. interjection (*ej; ah*).

The main feature of the above listed word (formal) classes is their division into *inflectional* and *non-inflectional* categories. While substantive words, adjective words and the verb constitute the former group, the adverb, predicate, preposition, conjunction, copula, and interjection share characteristics of the latter group. It is important to emphasize at this stage that inflection of the members of the Slovene word (formal) classes is carried out by the application of different endings, known also as *inflectional*

suffixes. Consequently, taking the direction towards the design of the automatic right-hand truncation—usually based on the list of endings—as the best way to achieve word conflation can again be supported. It is, therefore, mainly the inflectional group of words which will be described in the following sections. This decision can also be justified by the fact that most of the non-inflectional words (e.g., prepositions, conjunctions) belong to the so-called non-content bearing words and can, therefore, be considered as candidates for a list of stop-words in an information retrieval system.

To outline and illustrate the Slovene inflectional morphology, the concept of *grammatical categories* has to be introduced. By this concept, the general formal and semantic properties which bring together words of different concrete-lexical meanings into the same form-class are described (Lencek, 1982). The grammatical categories are inherent, as gender in substantives and aspect in verbs, or syntactically determined, as gender and number in adjectives. Together with their word (formal) classes they make up the paradigmatic system of Slovene morphology.

In general, Slovene shares its grammatical categories with other Slavic languages. It differs from them in that it possesses the category of dual in addition to singular and plural, and that its nominal system does not possess a special morphological form to express an appeal (vocative).

According to Lencek (1982), the basic grammatical categories of the Slovene nominal "parts of speech" are: gender, number, case, the animate/inanimate distinction in substantives, the definitive/indefinitive, and the positive/nonpositive oppositions in adjectives. The basic grammatical categories of the Slovene verbal forms are: aspect, voice, person, number (and marginally gender), tense, and mood. The presence or absence of some of these categories in an inflectional form makes the inflectional forms of Slovene conjugation finite and nonfinite; a finite form is marked by the category of person (and sometimes gender), a nonfinite form does not express person. In addition, the Slovene inflectional forms are either simple (e.g., the present tense, imperative) or compound (e.g., the past, future).

A "match" of these grammatical categories with their word (formal) classes is shown

Word class	Gender	Numb.	Case	Pers.	Degree	Aspect
noun	x	x	x			
verbal noun	x	x	x			
subs. pronoun	x	x	x	x		
adjective	x	x	x	x		
numeral	x	x	x			
adj. pronoun	x	x	x			
verb (pers.form)	x	x	x	x		
descr. partic.	x	x	x	x	x	x
adverb				x		
predicate				x		
preposition						
conjunction						
copula						
interjection						

Table 3.1: Word (formal) classes and grammatical categories

in Table 3.1 which illustrates the inflectional complexity of the Slovene language.

A "match" of the grammatical categories and word (formal) classes results in different *inflectional patterns*. For example, declension is a feature of all substantive and adjective word classes since in Slovene the relationship between words in sentences is expressed by the application of six cases.

Bearing in mind the design of a stemming algorithm for the Slovene language, it is important at this point again to emphasize that all inflected word forms in Slovene consist of two main parts: a stem and a suffix. While a *stem* can be defined as the content-bearing part of the inflected word, a *suffix* represents those units of the inflected word which mark its gender, number, case, person, etc. The following are some examples:

PERSON: *dela -m -š* – ... (*working*);
CASE: *lip -a -e -i* ... (*lime-tree*);
GENDER: *lep* – *-a -o* ... (*pretty*);
NUMBER: *lep* – *-a -i* ... (*pretty*).

It is evident that the employment of suffixes plays a major role in the inflectional morphology of the Slovene language. To explain how and to what extent this affects the development of a Slovene stemming algorithm, illustrations of the morphological

structure of the Slovene language will be given in the subsections below. The framework of this description will be based on the grammatical categories, starting with the category of gender.

3.3.3 The category of gender

The Slovene language distinguishes three genders: the masculine (*on - he*), the feminine (*ona - she*) and the neuter (*ono - it*). It is interesting to note that in the majority of the Slavic languages, gender is inherent in substantives, inflected in adjectives, and not expressed in pronouns (Lencek, 1982). Slovene, however, has extended gender to personal pronouns, and marginally to verbal inflection. The following are some examples of the application of gender in Slovene:

	Masculine	Feminine	Neuter	
NOUN	*brat - brother*	*sestra - sister*	*dete - baby*	
	stol - chair	*miza - table*	*mesto - town*	
	med - honey	*knjiga - book*	*sonce - sun*	
ADJECTIVE	*lep*	*lepa*	*lepo*	*pretty*
	slovenski	*slovenska*	*slovensko*	*Slovene*
	mlad	*mlada*	*mlado*	*young*
VERB (PARTICIPLE)	*delal*	*delala*	*delalo*	*worked*
	zaželen	*zaželena*	*zaželeno*	*desired*
	pil	*pila*	*pilo*	*drunk*
PRONOUN	*moj*	*moja*	*moje*	*my*
	on	*ona*	*ono*	*he, she, it*
	mi	*me*	*me*	*we*

As can be seen from the examples above—although there are some departures from the rules—words ending in the nominative case singular in *-a* are mostly feminine (*miza*), in *-o* and *-e* are mostly neuter (*dete, mesto*), and in a *consonant*, *-i* or *-u* are mostly masculine (*brat, slovenski*). The following two sentences illustrate the use of gender in Slovene and compare it to English:

Slovene	English
Tvoj novi prijatelj je raziskovalec.	*Your new friend is a researcher.*
*Tvoj**a** nov**a** prijateljic**a** je raziskovalk**a**.*	*Your new friend is a researcher.*

3.3.4 The category of number

The morphology of Slovene is unusual because besides the singular and plural, the *dual* is also used when referring to two persons or objects. The following are some examples of this:

Singular	Dual	Plural
*en***a** *miz***a** (*one table*)	*dv***e** *miz***i** (*two tables*)	*tr***i** *miz***e** (*three tables*)
*en***o** *mest***o** (*a town*)	*dv***e** *mest***i** (*two towns*)	*tr***i** *mest***a** (*three towns*)
lep (*pretty - M*)	*lep***a** (*pretty - M*)	*lep***i** (*pretty* - M)
*lep***a** (*pretty* - F)	*lep***i** (*pretty* - F)	*lep***e** (*pretty* - F)
on (*he*)	*onadva* (*they two*)	*oni* (*they*)

As illustrated above, the category of number is not only applied to nouns, but also to adjectives and pronouns. This feature distinguishes Slovene sharply from English since we know that the latter is characterized by singular and plural, and that the most common suffixes in the plural are either *-s* or *-es*.

3.3.5 The category of case

However, one of the most striking differences between Slovene and English morphology is the fact that in Slovene the inflection denotes not only the number form, but also the relationship of individual words in the sentence, which in English is expressed by the use of prepositions. These forms are called *cases*. There are six of them in the Slovene language: Nominative, Genitive, Dative, Accusative, Locative, Instrumental.

The category of case is relevant to the following word (formal) classes: nouns, verbal nouns, adjectives, and pronouns. The examples listed below illustrate the consequences of the application of the category of case—a phenomenon known as declension—and are mainly related to the increased number of new suffixes.

The declension of nouns follows the following four patterns mainly:

	Singular					Plural					
1. *miz-a*	*-e*	*-i*	*-o*	*-i*	*-o*	*miz-e*	–	*-am*	*-e*	*-ah*	*-ami*
2. *nit-*	*-i*	*-i*	–	*-i*	*-jo*	*nit-i*	*-i*	*-im*	*-i*	*-ih*	*-mi*

3.	*korak-*	*-a*	*-u*	*-*	*-u*	*-om*		*korak-i*	*-ov*	*-om*	*-e*	*-ih*	*-i*
4.	*mest-o*	*-a*	*-u*	*-o*	*-u*	*-o*		*mest-a*	*-*	*-om*	*-a*	*-ih*	*-i*

Dual

1.	*miz-i*	*-*	*-ama*	*-i*	*-ah*	*-ama*
2.	*nit-i*	*-i*	*-ima*	*-i*	*-ih*	*-ima*
3.	*korak-a*	*-ov*	*-oma*	*-a*	*-ih*	*-i*
4.	*mest-i*	*-*	*-oma*	*-i*	*-ih*	*-i*

The following sentences, using only the singular of the word *mesto* (*a town*), illustrate the use of cases in Slovene:

Slovene	English
1. *To je mest***o**.	*This is a town.*
2. *Ne vidim nobenega mest***a**.	*I can't see any town.*
3. *Približujem se mest***u**.	*I'm walking towards this town.*
4. *Kako bi opisal to mest***o**?	*How would you describe this town?*
5. *Kdo živi v tem mest***u**?	*Who lives in this town?*
6. *Pod tem mest***om** *teče reka.*	*There is a river beneath the town.*

The second group of word (formal) classes undergoing declension are adjectives; they are declined as follows:

	Singular					Plural					
M	*lep-*	*-ega*	*-emu*	*-/-ega*	*-em*	*-im*	*lep-i*	*-ih*	*-im*	*-e*	*-ih*	*-imi*
F	*lep-a*	*-e*	*-i*	*-o*	*-i*	*-o*	*lep-a*	*-ih*	*-im*	*-a*	*-ih*	*-imi*
N	*lep-o*	*-ega*	*-emu*	*-o/-ega*	*-em*	*-im*	*lep-a*	*-ih*	*-im*	*-a*	*-ih*	*-imi*

Dual

M	*lep-a*	*-ih*	*-ima*	*-a*	*-ih*	*-ima*
F	*lep-i*	*-ih*	*-ima*	*-i*	*-ih*	*-ima*
N	*lep-i*	*-ih*	*-ima*	*-i*	*-ih*	*-ima*

The remaining two categories, i.e., pronouns and numerals, are declined in the same way as adjectives. Since they and, in particular, pronouns can be defined as *function words* and, therefore, are considered as candidates for a stop-word list, their inflectional characteristics indicate that a list of stop-words in Slovene will be more comprehensive than one created for English. To illustrate this point, the following is an example of how the interrogative pronoun *kaj* (*what*) can be declined:

kaj česa čemu kaj čem čim (*what*)

3.3.6 The category of degree

The *gradation* of adjectives and adverbs in Slovene can be defined by a process that is formally quite similar to that in English. As in the English language, there are three degrees of comparison in Slovene: the positive (the non-compared fundamental form), the comparative and the superlative. In addition, both languages are characterized by two forms of comparison:

- the comparative is formed by means of the adverb *bolj* (*more*), and the superlative *najbolj* (*the most*), placed before the adjective or adverb:

  bel (white) *bolj bel* *najbolj bel*

- the comparative is formed by suffixes *-ejši*, *-ši*, and *-ji* (in English *-er*), the superlative by adding the prefix *naj-* to the comparative (in English by the article *the* and the suffix *-est*, e.g., *the biggest*). The following are some examples of this form of comparison:

star (old)	*starejši (older)*	*najstarejši (the oldest)*
dolg (long)	*daljši (longer)*	*najdaljši (the longest)*
nizek (low)	*nižji (lower)*	*najnižji (the lowest)*

However, it is necessary to stress that the gradation of adjectives and adverbs in Slovene belongs to the word-formation concept. Thus, terms created during the gradation can be defined as new word forms and are characterized by gender, number, case, etc. For example, *starejši* can be declined as any other adjective.

3.3.7 Grammatical categories of the verbal forms

The large amount of literature devoted to the Slovene verbal system (for comprehensive descriptions see Paternost, 1963; Lencek, 1966; Toporišič, 1984) indicates that the *verb* is at the heart of both the word-formation theory and morphology of the Slovene language. Since it is not the intention of this chapter to outline the Slovene language in detail, the description of the main verbal categories has only one simple aim, i.e., to

illustrate how the employment of different verbal forms can significantly increase the number of suffixes and therefore influence the design of the stemming algorithm for the Slovene language.

According to Toporišič (1975), the principal verbal forms in Slovene are the infinitive (ending in *-ti* or *-či*) and the present tense (ending in the 1st person singular in *-m*). How other verbal forms derive from them can again be shown using the word *delati* (*to work*):

1. infinitive: *dela***ti**;
2. supine: *delat*;
3. participle ending in *-l*: *dela***l**;
4. participle ending in *-n*: *dela***n**;
5. verbal noun: *dela***nje**;
6. present tense: *del***am**;
7. imperative: *del***aj**;
8. participle ending in *-č*: *delajo***č**.

The main grammatical categories of the verbal forms are person, tense, aspect and mood. The examples below illustrate how their employment affects the behaviour of the verb.

The category of person

Verbal forms are related to the three types of the category of person (1st, 2nd, 3rd person). The following is an example of the conjugation of the verb *delati* (*to work*):

	Singular	Dual	Plural
1st pers.	*dela* **-m**	*dela* **-va**	*dela* **-mo**
2nd pers.	**-š**	**-ta**	**-te**
3rd pers.	-	**-ta**	**-jo**

The category of tense

There are four tenses in Slovene. Except for the present tense, they are all formed with the participle ending in *-l* (or in *-n* or *-t*) and the auxiliary:

Present tense: *del***am** (*I work*);
Past tense: *del*a**l sem** (*I worked*);
Future tense: *del*a**l bom** (*I shall work*);
Pluperfect tense: *del*a**l sem bil** (*I had worked*).

The category of mood

There are three moods in Slovene:

Indicative: *del*a**m** (*I work*), *del*a**l sem** (*I worked*);
Imperative: *del*a**j** (*work*), *del*a**jmo** (*let us work*);
Conditional: *del*a**l bi** (*I would work*), *del*a**l bi bil** (*I would have worked*).

The category of aspect

Every verb obligatorily belongs to one of two classes of aspect: perfective or imperfective. Thus, the majority of Slovene verbs occur in two formal varieties, one of which implies that the action is understood as limited (a perfective verb, e.g., *to reach*), and the other that the action is understood as unlimited (an imperfective verb, e.g., *to be reaching*). The contrast between them is expressed not only by different suffixes, but also by a radical alternation of the stem; the latter is of particular concern to the design of the automatic word conflation algorithm for the Slovene language. The following are some examples of perfective verbs, followed by imperfective verbs; they illustrate both the appearance of new suffixes and alternations within stems:

*dvig***n**i*ti - dvig***a**ti*	*to lift - to be lifting*
*pre***ne**s*ti - pre***na**šati*	*to transfer - to be transferring*
*se***č**i - se***ga**ti*	*to reach - to be reaching*
*pri*ti - pri***ha**jati*	*to come - to be coming*

3.4 Types of morphemic alternations

As pointed out in previous sections, the characteristics of the inflectional morphology of Slovene, together with the concept of word-formation, constitute a starting-point for the design of the Slovene stemming algorithm. However, there is one additional feature of the Slovene language which has to be considered in the development of any word conflation procedure. This feature corresponds to the frequent *alternation* occurring in both stems and suffixes during the inflection of word forms. Whilst the rich inflectional morphology of the Slovene language indicates a need mainly for developing an extensive list of suffixes, the process of alternation not only causes new endings to be added to the suffix list, but also requires the introduction of context-sensitive and recoding rules as a part of the automatic word conflation procedure.

Although some types of alternation have already been noted in the examples illustrating the category of aspect (*pre***ne***sti* - *pre***na***šati*) and the category of degree (***niz****ek* - ***niž****ji*), the main aim of this section is to describe and illustrate concisely the basic types of modifications. Two main sources have been consulted in the preparation of this section, the first being Lencek (1982), and the second, Toporišič (1984).

According to Lencek (1982), there are three basic alternation types—prosodic, vocalic, and consonantal—common to both the nominal and verbal systems in Slovene. Since prosodic alternations mainly involve alternations of stress, they are not relevant to the written form of the Slovene language. Thus, the next subsections will be concerned with vocalic and consonantal modifications, occurring in both suffixes and stems.

3.4.1 Vocalic alternations

There are two types of vocalic alternation which are potentially interesting to the design of a stemming algorithm for Slovene:

- the *vowel* $\sim$ *zero* alternations;
- the *grave* $\sim$ *acute vowel* alternations of the $o \sim e$ type.

In the first type of alternation, there are four vowels, i.e., *e, i, o, a*, which alternate with *zero* in the nominal system of the Slovene language. A *zero* can be defined as a consonantal cluster, particularly of a *consonant + sonorant* type, occurring at the end of a stem.

The following are some examples of the *e* ~ *zero* alternation, occurring during the inflection of words:

veter - vetra	(*wind*)
dinozaver - dinozavra	(*dinosaur*)
bolezen - bolezni	(*illness*)
miren - mirnega	(*peaceful*)
sestra - sester	(*sister*)
brati - berem	(*to read*)

As far as longer word forms are concerned (for example, *dinozaver - dinozavra*) this type of modification should not represent a serious problem in the design of the stemming algorithm; a deletion of the suffixes *-er* or *-ra* can simply be employed, producing the stem *dinozav-*. However, a removal of these two endings from shorter terms, for example, *veter - vetra* would result in a stem *vet*. Consequently, this can cause a serious overstemming problem (consider a term *vet-o*). On the other hand, leaving both terms untouched would result in serious understemming. It is obvious that these examples simply indicate a need for introducing *recoding rules* as an inevitable part of the Slovene stemming algorithm.

The second type of alternation (i.e., the *o* ~ *e* type) can be described as follows. If a letter before a suffix beginning with *-o* is one of the consonants *c, č, ž, š, j*, then the grave vowel *-o* is automatically changed to the acute vowel *-e*. The following are two examples:

fantom (instr. case of *boy*) vs. *stricem* (instr. case of *uncle*)
dekletom (instr. case of *girl*) vs. *mladeničem* (instr. case of *youngster*)

As far as the design of the Slovene stemming algorithm is concerned, the main effect of this type of modification is again an increased number of endings becoming candidates for the suffix list.

3.4.2 Consonantal alternations

The two types of consonantal alternation in the Slovene language are known as *substitutive softenings* of Type I and Type II (Lencek, 1982).

The substitutive softenings of Type I, known also as $K \sim \check{C}$, are:

$k \sim \check{c}$	$t \sim \check{c}$	$sk \sim \check{s}\check{c}$	$r \sim rj$	$p \sim plj$
$g \sim \check{z}$	$c \sim \check{c}$	$st \sim \check{s}\check{c}$	$l \sim lj$	$b \sim blj$
$h \sim \check{s}$	$d \sim j$	$zd \sim \check{z}$	$n \sim nj$	$v \sim vlj$
	$s \sim \check{s}$	$sl \sim \check{s}lj$		$m \sim mlj$
	$z \sim \check{z}$			

It has to be stressed that the $K \sim \check{C}$ types of alternations are mainly found in verbal inflections; in addition, they also occur in the formation of comparatives. The following are some examples:

*jo**k**ati - jo**č**em*	*vi**s**ok - vi**š**ji*	*zgu**b**iti - zgu**bljen***
*str**g**ati - str**ž**em*	*ta**n**ek - ta**nj**ši*	*prelo**m**iti - prelo**mljen***
*prene**s**ti - prena**š**ati*	*dale**č**- da**l**je*	*preno**v**iti - preno**vljen***

The substitutive softenings of Type II, known also as $K \sim C$, have only two instances:

$k \sim c$
$g \sim z$

These alternations are also mainly a characteristic of the verbal inflection, particularly the imperative mood, e.g.:

re**k** (stem of *say*): re**c***i*, re**c***ite*;
le**g** (stem of *lay*): le**z***i*, le**z***ite*

In general, consonantal alternations are very frequent and thus can introduce additional requirements into the design of the stemming algorithm.

3.4.3 Truncation

In addition to vocalic and consonantal alternations, the Slovene verbal inflection is also characterized by truncation, which consists of the modification of a basic verbal stem (Lencek, 1982).

Truncation very often involves deeper changes in the stem. Thus, the vocalic stems ending in *-ov-a-*, when truncated, change their *-ov-* to *-uj-*. For example:

*razisk***ov-a**: *razisk***uj**, *razisk***uj***em*, *razisk***uj***eta*
*dar***ov-a**: *dar***uj**, *dar***uj***em*, *dar***uj***ejo*

Apart from the examples of the alternations outlined above, there are many more cases of morphemic modifications occurring in the Slovene inflectional system. They are concisely described in a highly systematic way by Toporišič (1984).

3.4.4 Complexity of Slovene morphology, using an example

On the basis of the discussion above about the morphological structure of the Slovene language, two main points can be emphasized:

- the Slovene language displays features of the extremely rich inflectional morphology in both verbal and nominal systems;
- in addition, Slovene is characterized by various types of morphemic alternations, occurring in both stems and suffixes during the inflection.

As described at the beginning of this chapter, we can now return to the stem RAZISKOVA- (*research*) and give a list of all its variants to illustrate the above two points:

RAZISKAVA	RAZISKOVANJ	RAZISKANI
RAZISKAVE	RAZISKOVANJEMA	RAZISKANO
RAZISKAVI	RAZISKOVANJIH	RAZISKUJEM
RAZISKAVO	RAZISKOVANJI	RAZISKUJEŠ
RAZISKAV	RAZISKOVALNI	RAZISKUJE
RAZISKAVAMA	RAZISKOVALNEGA	RAZISKUJEVA
RAZISKAVAH	RAZISKOVALNEMU	RAZISKUJETA

RAZISKAVAM
RAZISKAVAMI
RAZISKOVALEC
RAZISKOVALCA
RAZISKOVALCU
RAZISKOVALCEM
RAZISKOVALCEV
RAZISKOVALCEMA
RAZISKOVALCIH
RAZISKOVALCI
RAZISKOVALCE
RAZISKOVALKA
RAZISKOVALKE
RAZISKOVALKI
RAZISKOVALKO
RAZISKOVALK
RAZISKOVALKAMA
RAZISKOVALKAH
RAZISKOVALKAM
RAZISKOVALKAMI
RAZISKOVANJE
RAZISKOVANJA
RAZISKOVANJU
RAZISKOVANJEM
RAZISKOVANJI
RAZISKOVALNEM
RAZISKOVALNIM
RAZISKOVALNA
RAZISKOVALNIH
RAZISKOVALNIMA
RAZISKOVALNIMI
RAZISKOVALNE
RAZISKOVALNO
RAZISKOVATI
RAZISKATI
RAZISKAT
RAZISKAL
RAZISKALA
RAZISKALI
RAZISKAN
RAZISKANEGA
RAZISKANEMU
RAZISKANEM
RAZISKANIM
RAZISKANA
RAZISKANIH
RAZISKANIMA
RAZISKANIMI
RAZISKANE
RAZISKUJEMO
RAZISKUJEJO
RAZISKUJ
RAZISKUJVA
RAZISKUJTA
RAZISKUJMO
RAZISKUJTE
RAZIŠČI
RAZIŠČIVA
RAZIŠČITE
RAZIŠČIMO
RAZIŠČEJO
RAZISKUJOČ
RAZISKUJOČEGA
RAZISKUJOČEMU
RAZISKUJOČEM
RAZISKUJOČIM
RAZISKUJOČA
RAZISKUJOČIH
RAZISKUJOČIMA
RAZISKUJOČE
RAZISKUJOČIMI
RAZISKUJOČI
RAZISKUJOČO

In addition to 94 variants of the stem RAZISKOVA—as identified by the author—the following are some other examples, showing only a stem and number of its variants. These examples are taken from the experimental text corpus—as described in the next chapter—and, therefore, do not illustrate all possible forms of the particular stem.

Stem	Number of variants
RAZVOJ	*43*
UPORAB	41
INFOR	35
SPECIAL	26
SISTEM	25
STROK	24

3.5 Conclusions

Having listed a total of 94 variants for the stem RAZISKOVA- (*research*) and thus illustrating, using also some other words, the complexity of Slovene morphology, both in suffix variations and morphemic alternations, the following conclusions concerning further work on the stemming algorithm for the Slovene language can be drawn:

- Right-hand truncation, if properly designed, can play an enormously important role in improving both the effectiveness and efficiency of Slovene text retrieval systems;
- Morphemic alternations which very often cause deeper changes in a stem and are a frequent phenomenon in Slovene inflectional morphology, particularly in its verbal system, impose serious limitations on the idea of manual right-hand truncation;
- Bearing in mind other disadvantages of manual right-hand truncation, as described in previous chapters, the decision to design an automatic conflation procedure seems to be the most appropriate solution for the Slovene IR environment;
- Familiarity with the characteristics of the Slovene language indicates that the design of an automatic conflation procedure will involve the following factors:
    1. A list of stop-words will comprise a large number of terms.
    2. The best way of achieving automatic word conflation seems to be by developing a stemming algorithm, based on the longest-match principle. Trying to establish iteration patterns seems to be an extremely difficult and almost impossible task.
    3. To implement the longest-match principle, a list of suffixes is needed. There are indications that the list of endings will share some characteristics of the stop-word list, i.e., that of being very comprehensive.
    4. The stemming algorithm will necessarily require context-sensitive and recoding rules; the latter will be particularly important to avoid overstemming in

word forms having five or less characters.

5. However, the main aim of the design process will be to obtain a reasonable balance between, on the one hand, the number of rules, and on the other hand, simplicity and efficiency of processing.

Chapter 4

Development of a Stemming Algorithm for the Slovene Language

4.1 Introduction

4.1.1 Information retrieval research in Slovenia

When thinking about the development and design of a stemming algorithm for the Slovene language, considering in particular its morphological complexity, an interesting, but also contradictory situation comes to light. While, on the one hand, a certain degree of progress has been achieved in natural language processing research over the last ten years, information retrieval research, on the other hand, has taken a very small part in developing modern, non-conventional techniques to improve the effectiveness and efficiency of retrieval systems.

Research in natural language processing has been mainly carried out at the Institut Jozef Stefan in Ljubljana. The main aim of this research has been to develop natural language understanding concepts (Tancig, 1985). On the basis of a detailed analysis of the syntax of the Slovene language and using artificial intelligence methods, primary

interest has been focused in developing semantic schemes for the Slovene language. Although some of the research projects have contributed towards development in this area, there has been a significant lack of application of research results, particularly caused by the characteristics of Slovene morphology. In addition, there has so far been no link with the information retrieval research community for the potential transfer of research results to improve access to existing bibliographic databases.

From the 1970s onwards, a number of databases have been created in specialized information centres and libraries in Slovenia. At present, according to the report published by the Research Community of Slovenia (1989), there are 49 databases, 27 of which are bibliographic. It is interesting to note that the contents of the bibliographic databases are represented only by descriptors, and not by abstracts. With regard to programs, a number of different information retrieval systems are used, the most popular being the TRIP system, developed by PARALOG (1990) and employed on a VAX/VMS mainframe at the University of Ljubljana, and the ATLASS system, developed by the Institute for Information Science, University of Maribor (1990) and employed also on a VAX/VMS mainframe. However, all these systems share the characteristics of conventional retrieval systems, i.e., Boolean searching techniques are employed and professional intermediaries are needed to carry out on-line searches on behalf of end-users. Furthermore, the effectiveness and efficiency of these systems have rarely been evaluated. Consequently, modern, non-conventional methods and techniques of information retrieval, for example, automatic indexing, best-match searching, and term weighting, are neither incorporated into existing retrieval systems in Slovenia, nor has any research—with one exception described below—been carried out in this area.

However, there is no doubt that with developments in software technology, with the growing number of bibliographic and other types of databases in Slovenia and with increasing user demand for accurate and up-to-date information, the area of modern information retrieval research will become very attractive as an alternative to conventional, Boolean information retrieval. Therefore, the implementation of a Slovene language-based free-text retrieval system, particularly the design of a stemming algorithm which is the main scope of this PhD research project, together with results of

experimental research work carried out by Dimec (1988) as described below, serves as an important starting-point for the development of a new generation of statistically-based free-text retrieval systems for Slovene textual databases.

4.1.2 Computer analysis of the Slovene language in medicine

A computer analysis of the Slovene language in medicine (Dimec, 1988) has, so far, been the only published research report in Slovenia using statistically-based techniques for information retrieval. The main aim of the project was to find out the frequency and growth characteristics of Slovene free language in medicine for the potential employment of automatic indexing in medical retrieval systems.

Research experiments were carried out on a text corpus which consisted of Slovene medical articles, taken from scientific journals. More than 30,000 words were included in this corpus. Apart from testing a term discrimination model (as described by Salton *et al.*, 1975) and a two-Poisson distribution scheme (as described by Harter, 1975) as potential models for automatic term selection, a considerable amount of work was devoted to the compilation of both a stop-word list and a suffix list, particularly as a means of achieving dictionary compression in information retrieval.

On the basis of the frequency distribution of words from the text corpus, and of considering the characteristics of medical terminology, two extensive lists of stop-words were created. The first stop-word list (1,205 words) primarily included function words and number terms. The second stop-word list (2,866 words) consisted of terms which carried meaning but were thought not to be relevant to medical information retrieval. It is important to note that the second list included not only speciality words such as BOLEZEN (DISEASE), BOLNIK (PATIENT), MEDICINA (MEDICINE), but also other terms such as CENTIMETER (CENTIMETER), ČAS (TIME), DEFINICIJA (DEFINITION), NOVEMBER (NOVEMBER), GRAM (GRAMME), MESTO (TOWN), NAČRT (PLAN), OPIS (DESCRIPTION), etc. It is, therefore, not surprising that the employment of both stop-word lists resulted in a 72% compression of the existing text corpus. This level of compression is extremely high, since, for example,

van Rijsbergen (1979) reports about 30% – 50% compression where similar procedures were applied to an English language text. It is obvious that the main reason for this high compression can be found in the second stop-word list which included many words of potentially low value only for medicine, but not necessarily for the Slovene language in general. Unfortunately, there is no evidence in the research report by Dimec (1988) as to how the design of the second stop-word list, which goes drastically beyond a core of function words, would be balanced against the effectiveness of the medical information retrieval system if such a dictionary were employed in the medical database. The results of experiments in developing stop-word lists for the English language (see, for example, Jones and Bell, 1984) have shown that once a stop-word list includes more than just function words (i.e., prepositions, articles, pronouns, etc.), the selection of stop-words becomes a very subjective process since the potential list appears to be virtually boundless. Consequently, the effectiveness of the information retrieval system can be seriously threatened. Thus, the majority of stop-word lists, developed so far for the English language, include only function words, words from phrases and sometimes speciality words.

A simple conflation procedure was also developed as part of the research project by Dimec (1988) to avoid morphological variants of terms in the text corpus and to achieve additional compression. The conflation procedure uses a list of 381 suffixes which were generated from the reversed, alphabetically sorted list of words from the text corpus. The longest-match method was employed in word conflation; each suffix also having an associated rule for minimum stem length. Despite the complexity of Slovene morphology there are no recoding rules, and as noted by Dimec (1988), many improvements are necessary to achieve better word conflation; for example, words such as JETRA – JETER (nominative and genitive case of LIVER), or CELICA – CELIČNA (noun and adjective of CELL) did not match, although they should have matched. Both words are characterized by morphemic alternation of the $e \sim zero$ and $c \sim \check{c}$ type, as described in detail in Chapter 3.

Since there is no description of evaluation of this stemming algorithm—apart from testing the level of compression—Table 4.1 shows an example of the conflation results

when the procedure was applied to some variants of the familiar term RAZISKAVA.

Word (input)	Stem (output)	Word (input)	Stem (output)
RAZISKAN	RAZISK	RAZISKOVALNE	RAZISKOV
RAZISKAV	RAZISKAV	RAZISKOVALNEGA	RAZISKOV
RAZISKAVA	RAZISK	RAZISKOVALNEM	RAZISKOV
RAZISKAVAH	RAZISKAV	RAZISKOVALNEMU	RAZISKOV
RAZISKAVE	RAZISK	RAZISKOVALNI	RAZISKOV
RAZISKAVI	RAZISK	RAZISKOVALNIH	RAZISKOV
RAZISKAVO	RAZISKAV	RAZISKOVALNIM	RAZISKOV
RAZISKOVALCA	RAZISKOVALC	RAZISKOVALNO	RAZISKOV
RAZISKOVALCE	RAZISKOVAL	RAZISKOVANJ	RAZISKOVANJ
RAZISKOVALCEM	RAZISKOVALC	RAZISKOVANJA	RAZISKOV
RAZISKOVALCEV	RAZISKOVAL	RAZISKOVANJE	RAZISKOV
RAZISKOVALCI	RAZISKOVAL	RAZISKOVANJEM	RAZISKOVAN
RAZISKOVALCU	RAZISKOVALC	RAZISKOVANJU	RAZISKOV
RAZISKOVALEC	RAZISKOVAL	RAZISKUJE	RAZISK
RAZISKOVALNA	RAZISKOV	RAZISKUJEJO	RAZISKUJ

Table 4.1: Results of conflation of variants of the word RAZISKAVA, using the algorithm designed by Dimec (1988)

As can be seen from Table 4.1, many enhancements in the algorithm are needed to improve the results of word conflation. Reducing the total of 30 variants of the stem RAZISK- to 9 different forms, and obtaining only 5 terms (16.6%) with the common stem RAZISK- indicates a need for further work on the stemming algorithm. In addition, poor performance results correlate to the small compression achieved in this text corpus. Dimec (1988) reports about 12% reduction of the text corpus that remained after application of the stop-word lists. This is a very low figure when compared to the results reported by Lennon *et al.* (1981). These authors have evaluated various stemming algorithms on different test collections in the English language and the level of compression obtained ranged from 26.2% to 50.5%. The main reason for the low figure obtained by Dimec (1988) arises from the relatively small list of suffixes, which is not in accordance with the complexity of the morphological structure of the Slovene language.

There is no doubt that the results of this research project should be viewed as an important step towards the implementation of modern information retrieval methods and techniques in the Slovene textual database environment. However, more experimental work is needed in this area, particularly because some of the techniques so far developed and tested were based on a small text corpus and related only to medical terminology.

4.1.3 A general framework for the design of a stemming algorithm for the Slovene language

The most important objective of this project is the design of a powerful stemming algorithm that takes account of the language's morphological structure. On the basis of a review of different stemming algorithms, as described in Chapter 2, and of consideration of the complexity of Slovene morphology, as outlined in Chapter 3, the following framework of experimental research work was designed to achieve this objective. The research project was divided into two stages. The first stage of experimental work, based mainly on the so-called *frequency* approach, included the automatic compilation of a suffix list, development of a stop-word list and the design of a simple conflation procedure. Experiments were carried out on two different text collections, one consisting of terms from library and information science articles, the other covering the general area. In each database, almost 60,000 terms were included. In addition, an English text corpus, also covering the area of librarianship and information science, was used to illustrate language-dependent requirements in the design of the stemming algorithm.

The performance results of the frequency algorithm, the characteristics of the Slovene language, and some conclusions derived from work carried out by Dimec (1988), suggested an introduction to the second stage of the experimental research work, i.e., a need to develop more sophisticated methods and techniques in the stemming algorithm if the objective was to achieve better conflation results. However, the main aim of the experimental work in the second stage was to obtain a reasonable balance between, on the one hand, the number of rules, and on the other hand, simplicity and efficiency of

the conflation procedure.

Since the above outlined experiments were strictly applied in the design of the stemming algorithm, the course of the research project will be described in this chapter in the following order. First, the methodology employed in the experimental work will be outlined, followed by the analysis of the frequency distribution of words in the Slovene textual databases. On this basis the two main streams of the research work will be presented. Whilst in the first part the design and evaluation of both the stop-word list and a frequency algorithm will be outlined, the second part will describe the main points in the development and evaluation of the new stemming algorithm which will be incorporated into INSTRUCT.

4.2 A methodological framework of the experimental work

As pointed out above, there were two main objectives to be achieved as a result of the experimental research work:

- the production of the stop-word list;
- the design of the stemming algorithm, based on the list of suffixes.

Although there are many different approaches to the design of automatic conflation procedures, (see for example, Dawson, 1974; Field, 1975; Tarry, 1978; Hafer and Weiss, 1974), a similar approach to those used in the RADCOL project (Lowe *et al.*, 1973), the CITE system (Ulmschneider and Doszkocs, 1983), and in the computer analysis of the Slovene language in medicine (Dimec, 1988), was adopted as a first step in developing the stemming algorithm. Since this approach is based on the results of frequency distribution of words and suffixes in the textual databases, it can significantly help in the selection of terms to be included in the stop-word list, and provide a list of endings to be employed in the stemming algorithm.

The following are the major steps which are usually taken if the *frequency-based* approach is applied to words in the textual databases:

1. all words from the text corpus are extracted;
2. these words are ranked by frequency of their occurrence;
3. since the most frequent words are usually function words they are reviewed for their inclusion into the list of stop-words;
4. a list of stop-words is created; stop-words are removed from further analysis;
5. remaining terms are reversed and sorted into alphabetical order;
6. adjacent words in the ordered list are then compared and whenever a match of N characters is found, strings containing *1,2,...,N* characters are created;
7. these strings are sorted, cumulated, and the most frequent endings are either directly employed in the longest-match, context-free frequency algorithm or used as a starting-point for the manual selection of the suffix list.

There is one major advantage to the above described procedures, i.e., they can be almost completely automated and there is a very little need for any manual involvement. However, familiarity with the morphological structure of the Slovene language suggested the need for a certain degree of manual participation in both the development of the stop-word list and the design of the stemming algorithm. For example, since some of the word classes, in particular pronouns—although being defined as non-content bearing words—belong to the inflectional category of terms there is no guarantee that all their forms will always appear in the upper part of the term distribution table. In addition, a large number of Slovene terms undergo different types of morphemic alternation which affect both stems and endings during the inflection; it is difficult to note these changes automatically. Thus, all procedures used in developing a stop-word list and the new stemming algorithm will be a combination of both manual and automatic approaches.

All research experiments were carried out on words from two Slovene text corpora. The first corpus—referred to as KNJIŽNICA (LIBRARY in English)—consisted of terms from fourteen different articles on librarianship and information science. The articles covered topics such as library education in Slovenia, the use of serial publications

in university and special libraries, evaluation of library services and the use of new technology in libraries, *inter alia*. This corpus consisted of 59,088 word tokens.

The second corpus consisted of the word tokens that comprise Ciril Kosmač's novel POMLADNI DAN (A DAY IN SPRING), published in 1953, and transformed into machine-readable form in 1981 by Primož Jakopin. This machine-readable text was chosen since the novel is widely recognized as representing the Slovene language in all its beauty. The novel can thus be expected to exhibit the full range of the language's morphological complexity. It was assumed, therefore, that the initial list of both stop-words and suffixes would represent an important supplement to the lists produced from the words in the library and information science articles, i.e., KNJIŽNICA. The novel by Ciril Kosmač, which will be referred to as POMLADNI.DAN, comprises a total of 62,150 word tokens, a figure which is similar to the number of word tokens in KNJIŽNICA.

In addition, a third test collection was also employed in experiments, but this one consisted of English terms. The idea to use frequency-based techniques on an English text derived from a need to confront language-dependent requirements in the design process. The English text corpus which will be referred to as ENG.TEXT, consisted of 55,460 word tokens from a doctoral dissertation in the field of librarianship and information science (Ellis, 1987). It is thus comparable in size, in terms of the number of word tokens, with both of the Slovene corpora and is also comparable in subject matter with the KNJIŽNICA corpus.

4.3 Development of a stop-word list

4.3.1 Frequency distribution of terms

It is known from the early work on automatic indexing by Luhn (1958) that the frequency of occurrence of distinct words in natural language text has something to do with the importance of these words for purposes of content representation. Specifically, if all words were to occur randomly across the documents of a collection with equal frequen-

cies, it would be impossible to distinguish between them using quantitative criteria. Since it has been observed that the words occur in natural language text unevenly, they can be distinguished by their occurrence frequency. Or in other words, as noted by Luhn (1958):

> "The justification of measuring word significance by use-frequency is based on the fact that a writer normally repeats certain words as he advances or varies his arguments and as he elaborates on an aspect of a subject." (p. 160).

In fact, it is known that when the distinct words in a body of text are arranged in decreasing order of their frequency of occurrence (i.e., most frequent words first), the occurrence characteristics of the vocabulary can be characterized by the constant rank-frequency law of Zipf (1965) which is expressed in the following form:

$$rank \times frequency = constant$$

That is, the frequency of a given word multiplied by the rank order of that word will be approximately equal to the frequency of another word multiplied by its rank. The law has been explained by citing a general "principle of least effort" which makes it easier for an author to repeat certain words instead of coining new and different words. The least-effort principle also accounts for the fact that the most frequent words tend to be short function words (AND, OF, BUT, THE, etc.) which are easy to use in text.

Although *Zipf's Law* has been verified many times using text material across various subject areas and languages, it was again tested in this experimental work as a means of achieving two objectives:

- to design a stop-word list which could be used in information retrieval systems in Slovenia;
- to remove function words from both bodies of text in order to carry out automatic generation of suffixes from the remaining content-bearing words.

The results of the application of quantitative techniques on the words from both Slovene text collections, i.e., KNJIŽNICA and POMLADNI.DAN, are described below. First,

some general characteristics of the frequency distribution of the Slovene words are observed, followed by the testing of Zipf's Law. These results, together with the results obtained by the comparison of the English and Slovene text form a basis for discussion about the action to be taken in developing a list of Slovene stop-words.

The occurrence characteristics of terms

When all words from both Slovene text databases were extracted and ranked by frequency of occurrence, the following results were obtained. First, Table 4.2 clearly shows that both bodies of text, despite their coverage of different subject areas, produced a similar number of word types.

Text corpus	Word tokens		Word types	
	abs	%	abs	%
KNJIŽNICA	59,088	100	11,525	19.5
POMLADNI.DAN	62,150	100	10,988	17.7

Table 4.2: A comparison of the number of word types in databases POMLADNI.DAN and KNJIŽNICA

The reason for the slightly larger number of word types in the database KNJIŽNICA might be the fact that this corpus consists of articles written by different authors and covering various specialized areas. However, both bodies of text produced a reasonable number of distinct terms to be included in further experiments.

As expected, there was a large variation in the frequency of occurrence of the word types in text databases. If all word types are classified into 10 groups according to their decreasing order, i.e., the first group consisting of the 10% of the most frequent words, the second group comprising the next 10% of the most frequent words, etc., the following table (Table 4.3) illustrates the frequency distribution of terms in the text collections KNJIŽNICA and POMLADNI.DAN.

Word group (1,152 terms or 10%)	KNJIŽNICA		Word group (1,099 terms or 10%)	POMLADNI.DAN	
	abs	%		abs	%
1	39,749	67.3	1	45,463	73.1
2	5,992	10.2	2	4,907	7.9
3	3,500	6.0	3	2,853	4.5
4	2,309	3.9	4	2,198	3.5
5	1,778	3.1	5	1,234	2.0
6	1,152	1.9	6	1,099	1.8
7	1,152	1.9	7	1,099	1.8
8	1,152	1.9	8	1,099	1.8
9	1,152	1.9	9	1,099	1.8
10	1,152	1.9	10	1,099	1.8
Total	59,088	100.0	Total	62,150	100.0

Table 4.3: Frequency distribution of terms in KNJIŽNICA and POMLADNI.DAN, arranged in word groups in decreasing order

Table 4.3 shows that a very few word types provide a very high percentage of the observed tokens. Thus, the most frequent 10% of the word types in KNJIŽNICA account for 67.3% of the tokens whereas the bottom 50% account for only 9.5% of the tokens; the corresponding figures for POMLADNI.DAN are 73.1% and 9.0%. All terms in the bottom 50% are singletons, i.e., their frequency of occurrence is equal to 1. This type of frequency distribution is analogous to those observed in other languages.

The occurrence characteristics of the most frequent words in the Slovene language

Results of a detailed inspection of the group of the 10% most frequently occurring words from both Slovene bodies of text confirmed the expected results, i.e., terms in the very top of the listing were mainly function words. As an illustration, Table 4.4 displays a list of the 20 most frequently occurring words in the corpus KNJIŽNICA.

It can be seen from the Table 4.4 that the most frequent terms in the text corpus

Rank	Term	Frequency (abs)	Frequency (%)
1	IN	2,113	3.6
2	V	1,768	3.0
3	JE	1,270	2.1
4	ZA	905	1.5
5	NA	790	1.3
6	KI	688	1.2
7	DA	621	1.0
8	SE	608	1.0
9	SO	599	1.0
10	PA	544	0.9
11	TUDI	503	0.8
12	Z	440	0.7
13	S	385	0.6
14	KOT	333	0.6
15	KNJIŽNICE	331	0.6
16	O	319	0.5
17	NE	310	0.5
18	PO	292	0.5
19	ALI	290	0.5
20	PRI	271	0.5

Table 4.4: A list of the 20 most frequently occurring words in KNJIŽNICA

KNJIŽNICA are members of the following word (formal) classes: conjunction (IN; AND), preposition (V, ZA, NA; IN, FOR, ON) and auxiliary verb (JE, SO; IS, ARE). Since the main function of these words is in tying other words in sentences together, they are known as function words, common words or non-content bearing words. Function words are poor discriminators and cannot possibly be used by themselves to identify document content. Consequently, they are usually included in the stop-word list.

It is interesting to note that a content-bearing word, i.e., KNJIŽNICE (LIBRARIES), also appears in Table 4.4. Its high position derives from the fact that this list was created on the basis of the body of text describing the subject area of librarianship and information science. Apart from the word KNJIŽNICE, some other terms had a very high frequency of occurrence, for example INFORMACIJE (INFORMATION), achieving a rank equal to 30, and UPORABNIKI (USERS), having a rank equal to 35. Since

these terms carry a very low discrimination power in the specialized databases—in this case the library database—they are usually included in the stop-word list, and are known as so-called *speciality words.*

A detailed inspection of the most frequent terms in the database POMLADNI.DAN again confirmed expected results. As can be seen from Table 4.5, all of the 20 most frequent terms are non-content bearing words, thus indicating very little quantitative difference between terms in the two different subject areas.

Rank	Term	Frequency (abs)	Frequency (%)
1	JE	3,769	6.1
2	IN	2,453	3.9
3	SE	1,946	3.1
4	V	1,219	2.0
5	SEM	976	1.6
6	DA	880	1.4
7	PA	710	1.4
8	NE	653	1.1
9	NA	636	1.0
10	Z	575	0.9
11	KI	514	0.8
12	SO	503	0.8
13	BI	497	0.8
14	PO	442	0.7
15	ŠE	393	0.6
16	GA	392	0.6
17	S	382	0.6
18	NI	343	0.5
19	ZA	342	0.5
20	TAKO	334	0.5

Table 4.5: A list of the 20 most frequently occurring words in POMLADNI.DAN

In Table 4.5, quantitative characteristics of the function words are of the particular evidence. The most frequent 5 terms (JE, IN, SE, V, SEM; IS, AND, WILL, IN, AM), representing only 0.04% of all distinct words in the text body, account for 16.7% of term usage; on the other hand, as is evident from Table 4.3, the lowest 70% of the individual terms explain only 14.5% of term usage.

Since the frequency of term occurrence in both Slovene text collections confirmed a general "principle of least effort", it was assumed that the occurrence characteristics of both vocabularies would also correspond to the constant rank-frequency law of Zipf (1965).

Zipf's Law and the Slovene language

As already explained, Zipf's Law states that the product of the frequency of use of words and the rank order is approximately constant, i.e., $F \times R = C$.

Table 4.6 and Table 4.7 illustrate the results of the application of Zipf's Law to randomly selected words in the text collections KNJIŽNICA, and POMLADNI.DAN respectively.

Term	Rank (R)	Frequency (F)	Product of rank and frequency ($R \times F = C$)
PA	10	544	5,440
PRI	20	271	5,420
INFORMACIJ	30	182	5,460
VSE	40	130	5,200
DEJAVNOSTI	50	109	5,450
NISO	100	59	5,900
NOVE	200	33	6,600
PROBLEMOV	300	24	7,200
DOSLEJ	400	18	7,200
ČASA	500	16	8,000
DANAŠNJEM	1,000	8	8,000
GRADNJI	2,000	4	8,000
PRAVNO	3,000	3	9,000
MODULOM	4,000	2	8,000
VARSTVENIH	5,000	2	10,000
SLOVENJ	10,000	1	10,000
ŽIVO	11,525	1	11,525

Table 4.6: Results of the Zipf's Law on the words from the text collection KNJIŽNICA

Although Zipf's Law can be more or less confirmed using the frequency distribution

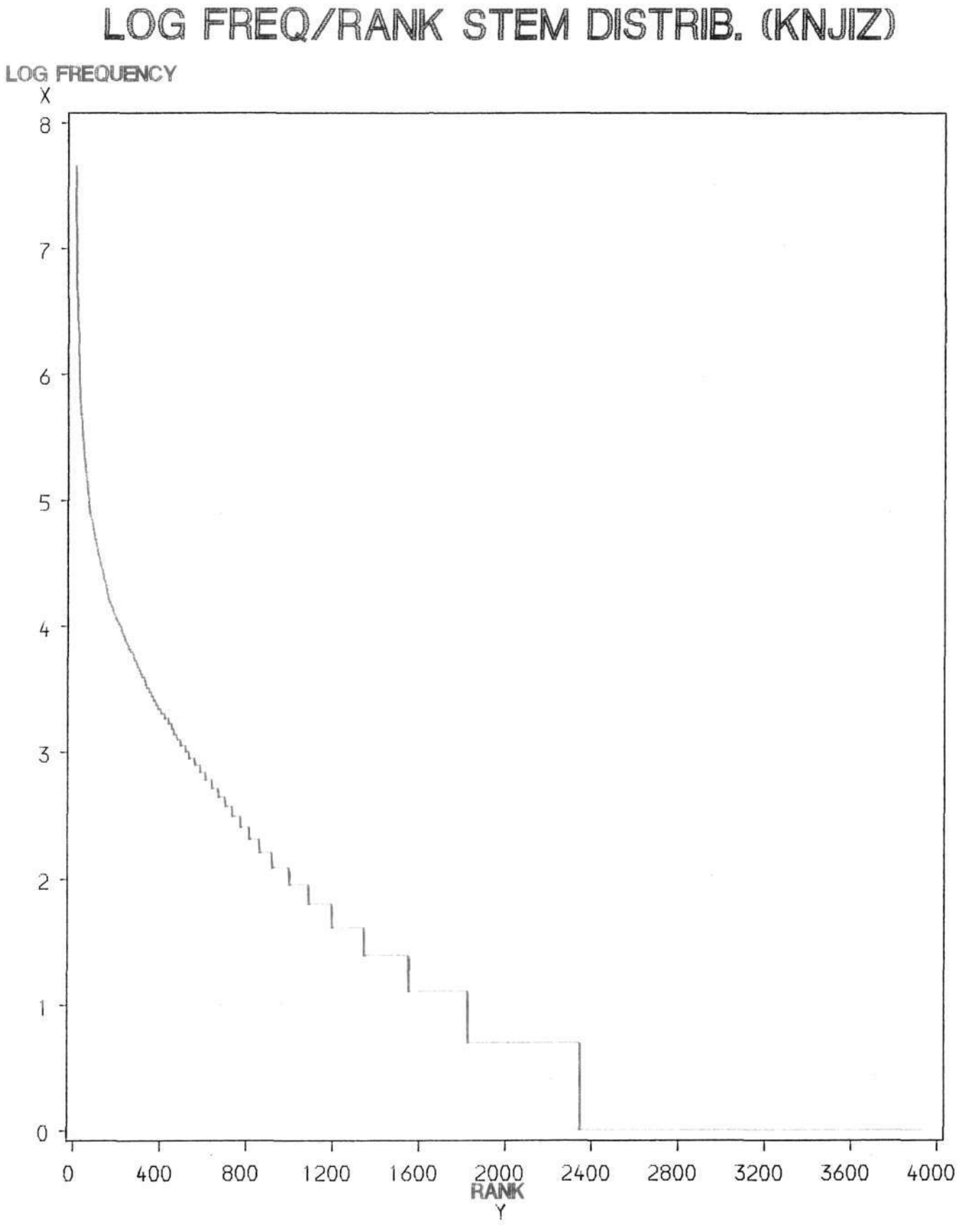

FIGURE 4.1
Plot of rank versus log of stem frequency

of words in the database KNJIŽNICA, it is interesting to note that the value of the constant (C) increases with decreasing frequency of words; this is especially evident for terms in the bottom part of Table 4.6, possessing frequencies of occurrence of 2 or less. This is in accordance with Booth's Law (Booth, 1967) which holds for words of very low frequency of occurrence and not for those of high frequency. Booth's Law derives from the fact that when the complete word frequency count is made for a text, words of high rank—that is of low frequency—occur in such a way that many words have the same frequency. Thus, the less frequently occurring terms show considerable departure from Zipf's Law.

This phenomenon may again be explained by the morphological structure of the Slovene language which divides the original stem into a large number of variants, thus dispersing the frequency of the stem occurrence over its variants. For example, in the text corpus KNJIŽNICA, a total of 5,235 terms out of 11,525 word types, i.e., 45.4%, occurred with term frequency equal to 1. A similar situation also happened in the text corpus POMLADNI.DAN, as shown in Table 4.7.

Using Zipf's Law, it is possible to produce curves of the form as used by Luhn (1958) to define significant words in the document collection, or in other words, to exclude non-relevant terms. Figure 4.1 represents a plot of the logarithm of frequency of terms in the database KNJIŽNICA against their ranked order of frequency and, therefore, illustrates how such a curve characterizes the Slovene language. It has to be emphasized that in order to maintain exactly the same approach as described by Luhn (1958), the frequency of words used in the figure in fact represents the frequency of already conflated words, using the new Slovene stemming algorithm which will be described at the end of this chapter.

The course of the curve in Figure 4.1 is very similar to those produced for various bodies of text, mainly for the English language (see, for example, Luhn, 1958; van Rijsbergen, 1979; Ashford and Willett, 1988). According to Luhn (1958), two cutoffs, i.e., the first one for the high-frequency terms, and the second one for the low-frequency words, can be defined in such a diagram in order to remove them from the document

Term	Rank (r)	Frequency (f)	Product of rank and frequency ($R \times F = C$)
Z	10	575	5,750
TAKO	20	334	6,680
MI	30	239	7,170
KO	40	196	7,840
KER	50	177	8,850
POČASI	100	68	6,800
BOMO	200	30	6,000
RADA	300	22	6,600
ČRNO	400	16	6,400
TEH	500	14	7,000
OBREKARJEV	1,000	7	7,000
ŠOLI	2,000	4	8,000
DOMAČEGA	3,000	2	6,000
SLEKLA	4,000	2	8,000
DELAVČEK	5,000	1	5,000
VIHRAL	10,000	1	10,000
ŽVIŽGANJE	10,988	1	10,988

Table 4.7: Results of the Zipf's Law on the words from the text collection POMLADNI.DAN

collection. Consequently, as concluded by Luhn (1958), since neither high- nor low-frequency terms are good content identifiers, the remaining medium-frequency words can be used to identify relevant terms.

For the purpose of developing a stop-word list to be used in Slovene information retrieval systems, the notion of Luhn (1958) concerning the high-frequency terms is of a particular interest. As evident from the results of the quantitative analysis of both KNJIŽNICA and POMLADNI.DAN, it is theoretically possible to create a list of Slovene stop-words, using the *frequency-based* approach. Indeed, as described below, information about the frequency distribution of Slovene terms served as one of the starting-points in the design of the stop-word list. However, the complex morphological structure of the Slovene language points out that additional manual involvement in the design process is inevitable. This notion is clarified in the section below where the main

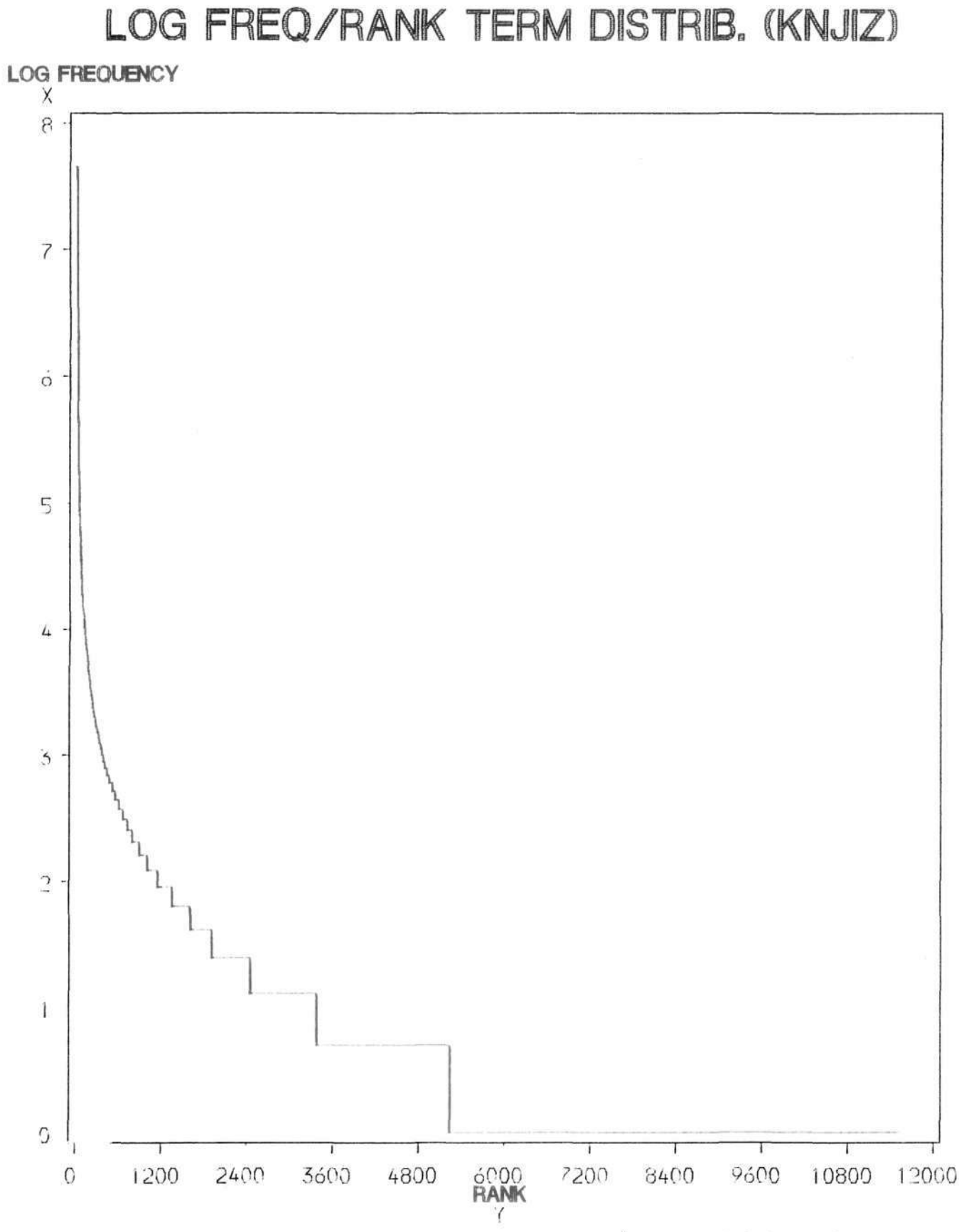

FIGURE 4.2
Plot of rank versus log of term frequency

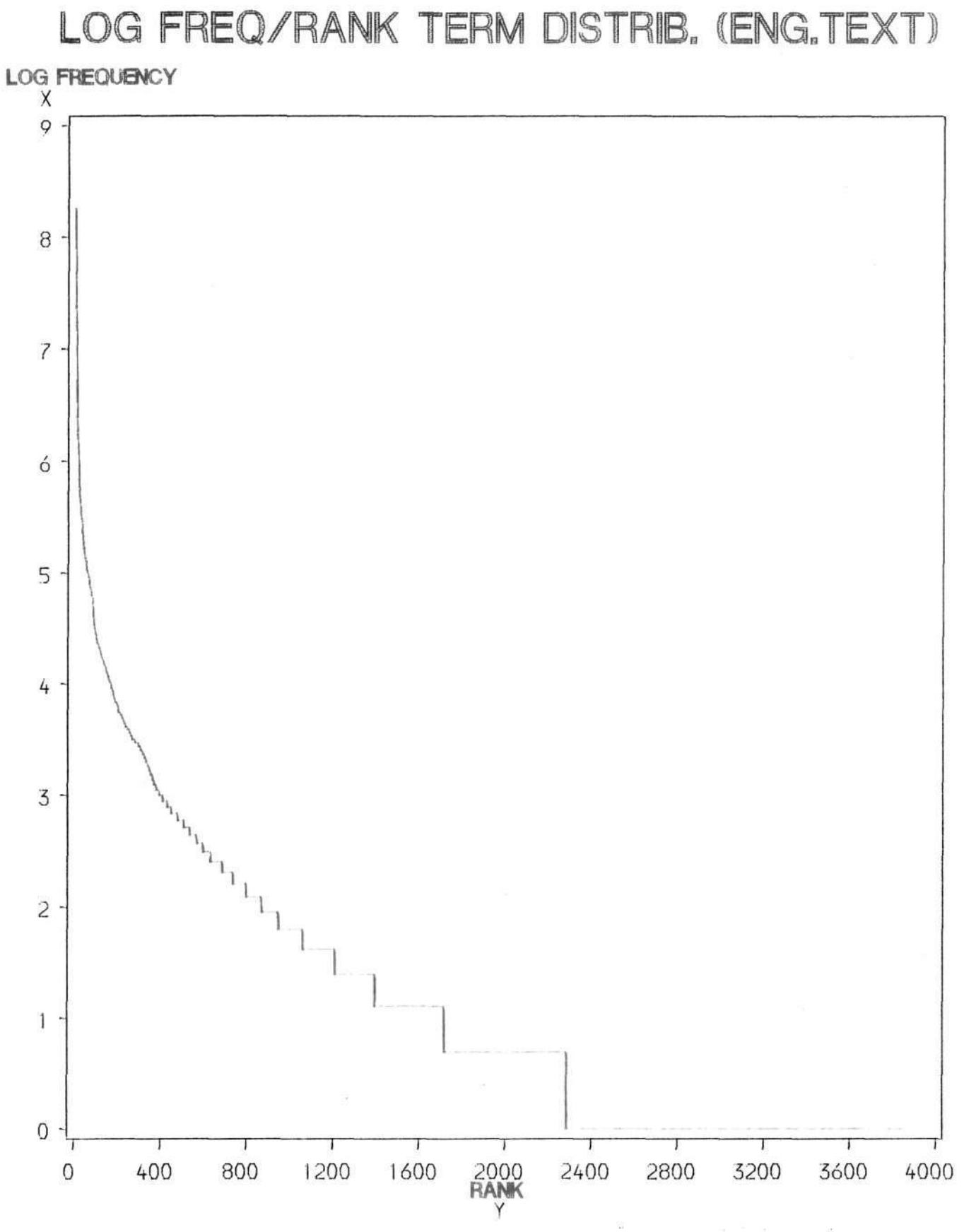

FIGURE 4.3
Plot of rank versus log of term frequency

quantitative differences between the Slovene and English language are described.

Quantitative comparison of the Slovene and English language

Word types from two text collections, the first one consisting of the Slovene terms (KNJIŽNICA), and the second one comprising English words (ENG.TEXT), were used as the basis for the quantitative comparison between these two languages.

Text corpus	Word tokens		Word types	
	abs	%	abs	%
KNJIŽNICA	59,088	100	11,525	19.5
ENG.TEXT	55,460	100	3,868	7.0

Table 4.8: A comparison of the number of word types in databases KNJIŽNICA and ENG.TEXT

Table 4.8 clearly demonstrates that both databases, despite having similar dictionary size and covering the same subject area, produced completely different totals of word types. While, on the one hand, the Slovene text corpus consisted of 11,525 word types, i.e., 19.5% of the total number of term occurrences, the English database contained only 3,868 word types, i.e., only 7.0% of the total number of term occurrences. The main reason behind this striking difference is the morphology of both languages, of which Slovene is by far the more complex, as illustrated in Chapter 3.

Similarly, a plot of the logarithm of the frequency of word types against their ranked order of frequency, applied to both languages separately, yields two different hyperbolic curves, as shown in Figures 4.2 and 4.3. It is interesting to note that the frequency distribution of terms in the Slovene text creates a more concave curve (see Figure 4.2) than the one produced for words in the English text (see Figure 4.3). The explanation can again be found in the inflectional morphology of the Slovene language. On the one hand, some members of word classes (prepositions, conjunctions)

are not inflected in the sentences, and these individual terms can therefore potentially reach a high frequency of occurrence whereas the inflected members of the word classes (nouns, adjectives, pronouns, etc.) have, on the other hand, theoretically less chance of achieving high frequency since the original stems are split during inflection into a large number of various terms, thus a potentially high-frequency of a given stem is dispersed among these variants. Whilst, for example, a non-inflected term ALI (OR) achieves a frequency of occurrence 290, an inflected word ZBIRKA (COLLECTION) appears in different variations achieving different frequencies: ZBIRK (79), ZBIRKE (53), ZBIRKA (16), ZBIRKO (16), ZBIRKI (10), ZBIRKAH (7), ZBIRKAMI (7), ZBIRKAM (2). Consequently, the Slovene language is characterized both by the small amount of high-frequency words which account for a large percentage of term usage, and by a large number of low-frequency terms.

If all Slovene function words were included in the high-frequency terms there would be no difficulty in deciding upon the list of stop-words for information retrieval systems. A simple frequency-based approach would be used, extracting the most frequent words from the text and transferring them to the stop-list. However, some of the function words (pronouns, auxiliary verbs, etc.) are also characterized by their inflection; thus, no guarantee can be given that all their variants will appear in the list of the most frequent words. Examples of inflectional variants of non-content bearing words that occur only once in KNJIŽNICA include KAK (ANY), KATERI (WHICH), MOJ (MY), MOJEGA (MINE), NAM (US), NEKI (CERTAIN), TVOJ (YOURS), STE (ARE). The production of a stop-word list for the Slovene language thus entails a much greater level of detailed, manual involvement than is required for the construction of a stop-word list for the English language.

4.3.2 Design of the Slovene stop-word list

In order to achieve one of the main goals, i.e., to create a general purpose stop-word list for Slovene, potential candidates were extracted from the following three sources:

- a textbook, written by Toporišič (1984), in which Slovene grammar is described in a highly detailed and structured way;
- both Slovene text collections, i.e., KNJIŽNICA and POMLADNI.DAN;
- a list of stop-words, created by Dimec (1988).

The sections below give a brief description of how these sources were used in the construction of a stop-word list.

Slovene grammar by Toporišič (1984)

A textbook about Slovene grammar (Toporišič, 1984), in particular the section on Slovene morphology, has proved to be an extremely valuable source in developing a stop-word list. Its employment was especially beneficial for:

- defining a criterion for selection of stop-words because of its excellent description of the main word (formal) classes;
- selecting candidates for the stop-word list because word (formal) classes were illustrated in the book with many examples.

On the basis of this textbook, the following word (formal) classes were selected for inclusion in the dictionary of non-content bearing words: substantive pronoun, numeral, adjective pronoun, auxiliary verb, adverb, predicate, preposition, conjunction, and copula.

Since many examples were used in Toporišič (1984) to illustrate the above word (formal) classes it appeared to be theoretically possible to create a preliminary list of stop-words. Thus, a large number of terms, belonging to the above classes, were simply extracted from Toporišič (1984) and included in a dictionary. If a certain term was a member of the inflectional category, then all its possible variants were produced, using, for example, declension, gradation, conjugation, etc. When all these terms were merged into one file, a total of 1,059 distinct stop-words was produced.

A comparison of terms from this dictionary with terms created using the other two sources, i.e., both Slovene test collections and the stop-list developed by Dimec (1988), confirms the high quality of the description of Slovene morphology by Toporišič (1984). A decision to use Toporišič (1984) as a starting point in developing a stop-word list has, therefore, proved to be correct mainly because:

- many new terms were discovered which did not occur in the other two sources (for example, BODIMO, BOSTA, MARSIKAKŠNEMU, MOJIMA, NAJINEGA, ČIGAR, TISTIMA, etc.);
- a theoretical background was firmly established for the further selection and evaluation of non-content bearing words from the other two sources.

A selection of stop-words from the Slovene text corpora

It has already been emphasized that characteristics of the Slovene language enable, to a certain extent, the frequency approach to be employed in the design of the stop-word list. However, since some of the non-content bearing words belong to the inflectional category, not all candidates for the negative dictionary occur among the most frequent words. Thus, a manual review of all extracted words from test collections KNJIŽNICA and POMLADNI.DAN was inevitable in order to create a useful dictionary of stop-words.

The combination of the automatic frequency approach and the manual selection of stop-words produced the following total of candidates for the inclusion in the preliminary stop-list:

- KNJIŽNICA: among 11,525 word types, 931 terms were found to share characteristics of the stop-words;
- POMLADNI.DAN: among 10,988 word types, 792 terms were considered as candidates for the stop-word list.

A stop-word list, created by Dimec (1988)

As has already been described in the introductory section, Dimec (1988) created two lists of stop-words as a part of his project on the computer analysis of the Slovene language in medicine. While the first list consisted of non-content bearing words (mainly function words and number terms), the second list comprised meaningful terms which were assessed as not being relevant to the medical language.

Following the main objectives of our research project, the interest in the design of the new stop-word list focused only on the first dictionary created by Dimec (1988), which consisted of 1,205 distinct terms. A detailed examination confirmed the quality of this list, and thus almost all stop-words were considered for their incorporation into the final list. There was only one exception, i.e., numerals. Since Dimec (1988) included in his list a large number of numerals without being consistent, a decision was made to consider only some basic forms of numerals which occurred frequently in both Slovene text databases. This decision served as a prevention against too comprehensive a stop-list if all forms of all numerals were included.

A design of the final stop-word list

Having produced four preliminary stop-word lists, i.e., extracting non-content bearing terms from Toporišič (1984), selecting terms from two text corpora, and considering terms from the first stop-word list created by Dimec (1988), it was possible simply by merging them into one file to construct the final stop-word list. Such an approach resulted in a list consisting of 1,593 individual terms. To justify the employment of different sources and the use of both the manual and automatic involvement into the design process it is perhaps interesting to note that a total of 623 terms not existing in the first stop-word list created by Dimec (1988) was included in the final dictionary.

The final list of the Slovene stop-words is presented in a machine-readable form on a floppy disk which can be found at the end of this thesis. The decision to use such a presentation was caused by the comprehensiveness of the stop-word list. The same decision was applied also to the list of suffixes.

The following is a brief description of the main criteria which were used in the selection of stop-words:

- a stop-word is defined as a non-content bearing word;
- consequently, the members of the word classes, carrying low meaning were primarily considered as candidates for the negative dictionary; these terms are mainly function words, and belong to prepositions, pronouns, auxiliary verbs, conjunctions, etc.;
- in addition, a small core of other types of terms is also included in the list:
  - a limited number of numerals, i.e., basic forms of numerals ENA (ONE) and DVA (TWO);
  - a limited number of words occurring frequently in phrases, for example, V ZAČETKU (IN THE BEGINNING), Z VIDIKA (POINT OF VIEW), VKLJUČNO (INCLUDING), etc.;
  - a small core of verbs which also appear in phrases, for example, KAŽE (IT SHOWS), POVE (IT SAYS), SPADA (IT BELONGS), etc.
  - some other terms, carrying extremely low meaning in sentences, for example, DOLOČEN (CERTAIN), NASLEDNJI (NEXT), OSTALI (OTHERS), PREJŠNJI (PREVIOUS), etc.

Although the decision to include other general words with fairly low, but variable, semantic contents was often considered in the design process, for instance the words MOŽEN (POSSIBLE), POMENI (IT MEANS), IZREDNO (EXTRAORDINARY), POMEMBNO (IMPORTANT), etc., the definition of developing a general purpose stop-word list for the Slovene language led to the decision not to extend the stop-list beyond a core of function words, basic numerals, and words from phrases. Moreover, the final list does not include any speciality words since the negative dictionary was not designed with any particular textual database in mind.

It should be noted that there are some words in Slovene that have exactly the same written form but differ in their meaning, i.e., they are homographs, and also in

their pronunciation, owing to the difference between vowels in the written form and in their vocalic sounds; for example MED can mean either BETWEEN or HONEY, VAS either VILLAGE or the accusative of YOU, and MORALA either MORALE or (SHE) SHOULD. No such words were included in the stop-word list since it is impossible to distinguish between the variants without extensive semantic processing.

4.3.3 Evaluation of the new stop-word list

The evaluation of the new Slovene stop-word list was carried out on a sample of 10 abstracts (referred to as SLOV) from the articles stored in the corpus KNJIŽNICA. Bearing in mind the two main disadvantages of such an sample, i.e., a relatively small test collection (958 terms; 6,262 characters) and its library content (i.e., the appearance of more or less the same terms as used in the text corpus KNJIŽNICA which was one of the main sources in the design of the stop-word list), the results of this evaluation should be considered as a preliminary stage towards the evaluation on a much larger scale. Such an evaluation will be carried out on the basis of incorporation of both the stop-word list and the stemming algorithm into INSTRUCT.

The main aim of the preliminary stage of evaluation was to test the ability of the negative dictionary to achieve a reasonable compression in a body of text consisting of 958 words. The employment of the elimination technique in which the words from abstracts were compared with a stored stop-word list, resulted in a total of 598 remaining terms, or 62.4%. If the level of compression is expressed in terms of the number of reduced words, then 37.6% compression was achieved. This level of reduction varied from abstract to abstract, ranging from 31.9% to 43.7%.

If the level of compression is estimated in terms of the reduced number of characters in the text corpus, then the amount of reduction of the Slovene test collection drops to 20.2%; the elimination technique meant that the total of 6,262 characters was cut down to 5,002 characters, or 79.8%. The main reason for the lower percentage of reduction can be found in the length of the most frequent stop-words which rarely exceeds four characters.

These results are comparable to the levels of compression accomplished when similar procedures were applied to the English text (see, for example, van Rijsbergen, 1979). In order to prove this similarity, the English language-based equivalent of the above sample was produced (referred to as ENGL) and then processed by the English list of stop-words. This list consists of 294 terms and actually corresponds to the list as implemented within the INSTRUCT package. Table 4.9 shows the levels of compression that were achieved after the employment of the English negative dictionary; a comparison with the results obtained by the application of the Slovene stop-word list is also made.

Text	Terms	Non-deleted terms	Compression (%)	Characters	Non-deleted characters	Compression (%)
SLOV	958	598	37.6	6,262	5,002	20.2
ENGL	978	559	42.9	5,482	4,272	22.1

Table 4.9: The levels of compression achieved by the application of the Slovene and English negative dictionaries

As can be seen in Table 4.9, both stop-word lists produced similar levels of compression. The reason for the slightly larger number of deleted English terms may be found in the grammar; for example, articles (A, AN, and THE) are not used in the Slovene language at all.

More importantly, an inspection of the sets of words resulting from the use of the Slovene stop-word list showed that a successful level of indexing had been achieved. For example, consider the following abstract from the KNJIŽNICA corpus, which contains ninety-four words and 609 characters:

UPORABNIKI IN ONLINE JAVNO DOSTOPNI KATALOG

Jože Kokole

Predstavljen je fenomen online javno dostopnega kataloga oziroma s kratico OPAC (po angleškem online public access catalogue) pri računalniško podprtem poslovanju knjižnic oziroma knjižničnih sistemov, njegovega nastanka, razvoja in stanja v razvitih sredinah, njegovih načel,

karakteristik in pojavnih oblik. Obdelano je še: uporaba OPAC-ov prve in druge generacije v posameznih knjižnicah in v vzajemnih katalogih, odnos do online bibliografskih servisov, problemi končnih uporabnikov in uporabe OPAC-a, zahteve in pogoji za oblikovanje učinkovitega in uporabniško prijaznega iskalnega dialoga, perspektive za uvajanje OPAC katalogov pri nas.

After application of the stop-word list, and reduction to single case, the abstract now contains just sixty-four words and 500 characters (upper-case denotes processed text):

UPORABNIKI ONLINE JAVNO DOSTOPNI KATALOG
JOŽE KOKOLE

PREDSTAVLJEN FENOMEN ONLINE JAVNO DOSTOPNEGA KATALOGA KRATICO OPAC ANGLEŠKEM ONLINE PUBLIC ACCESS CATALOGUE RAČUNALNIŠKO PODPRTEM POSLOVANJU KNJIŽNIC KNJIŽNIČNIH SISTEMOV NASTANKA RAZVOJA STANJA RAZVITIH SREDINAH NAČEL KARAKTERISTIK POJAVNIH OBLIK OBDELANO UPORABA OPAC-OV GENERACIJE KNJIŽNICAH VZAJEMNIH KATALOGIH ODNOS ONLINE BIBLIOGRAFSKIH SERVISOV PROBLEMI KONČNIH UPORABNIKOV UPORABE OPAC ZAHTEVE POGOJI OBLIKOVANJE UČINKOVITEGA UPORABNIŠKO PRIJAZNEGA ISKALNEGA DIALOGA PERSPEKTIVE UVAJANJE OPAC KATALOGOV

These first evaluation results indicate that the decision to use three different sources in the design of the stop-word list has proved to be correct. Both the level of achieved compression and the actual removal of non-content bearing terms are quite encouraging evidence of the quality of the Slovene negative dictionary. Despite the fact that evaluation on a much larger scale is needed, the results above demonstrate that this list can successfully be used in Slovene information retrieval systems, and, since being domain-independent, employed in any textual database.

At this point it is important to emphasize that in non-conventional retrieval systems the removal of stop-words from the textual databases is not an isolated action but is usually followed by automatic word conflation.

4.4 Design of a stemming algorithm

The complexity of Slovene morphology suggests that it would be extremely difficult to develop an effective iterative stemming algorithm. Accordingly, the use of longest-match algorithms has been studied.

4.4.1 Development of a suffix list

Although it is in theory possible to design a list of endings for stemming purposes on the basis of prior experience of the language, the availability of a large number of word usages in both text collections, i.e., KNJIŽNICA and POMLADNI.DAN, suggested another approach: to develop the suffix list using the information about suffixes contained implicitly in the word usage in the text collection.

It has already been shown that both text corpora contained a large number of word types. Even the application of the new stop-word list to terms in both collections did not significantly reduce the total number of distinct words; in the text corpus KNJIŽNICA a total of 10,711 distinct terms remained (i.e., 814 terms were removed), and in the text corpus POMLADNI.DAN a total of 10,215 words was left (i.e., 773 terms were excluded).

Using the remaining distinct terms from both collections it is possible to design a method for the automatic generation of the suffix list. Usually, the low-frequency words are excluded from such a procedure since they consist of a certain amount of foreign names, proper names, and misspellings (see, for example, Lowe *et al.*, 1973). Although such terms occurred among low-frequency terms in the Slovene corpora, they were not removed from further analysis because inflectional morphology produced a large number of relevant terms also having low frequencies. Thus, all distinct terms which remained in the text collections after employment of the stop-word list were used for the automatic generation of suffixes.

Automatic generation of endings was carried out separately on both text corpora. The following routines were written to produce a list of suffixes:

- a procedure to create a list of reversed words, sorted into alphabetical order;
- a procedure to compare the initial characters of adjacent words in the list;
- a procedure to merge and sort suffixes by decreasing frequency of occurrence.

Therefore, the starting-point for the automatic generation of suffixes was the production of a list of word reversals. This list showed for each word, the word reversed, the word itself, and its frequency of occurrence; part of the resulting list is shown in Table 4.10.

Reversed Word	Word	Frequency
EJNAVOKSIZAR	RAZISKOVANJE	4
EJNAVOLED	DELOVANJE	27
EJNAVOLEDOS	SODELOVANJE	12
EJNAVOLSOP	POSLOVANJE	13
EJNAVOLSOPAZ	ZAPOSLOVANJE	5
EJNAVONEMI	IMENOVANJE	1
EJNAVORAV	VAROVANJE	10
EJNAVORAVAZ	ZAVAROVANJE	1
EJNAVORDAK	KADROVANJE	3
EJNAVOSIPDERP	PREDPISOVANJE	1
EJNAVOSIPO	OPISOVANJE	1
EJNAVOTEVSOP	POSVETOVANJE	1
EJNAVOTRČAN	NAČRTOVANJE	11
EJNAVOZEVOP	POVEZOVANJE	30
EJNAŠANBO	OBNAŠANJE	2
EJNAŠANV	VNAŠANJE	1
EJNAŠARPV	VPRAŠANJE	34

Table 4.10: Words and their reversed forms.

As can be seen in Table 4.10, reversed words are arranged in alphabetical order; thus, words sharing a common suffix, such as NAČRTOVANJE and POSVETOVANJE, appear together in the list.

On this basis, adjacent words in the ordered list were compared and whenever a match of N characters was found, strings containing *1, 2, ..., N* characters were created. Thus, the words NAČRTOVANJE and POSVETOVANJE would produce the

strings -E, -JE, -NJE, -ANJE, -VANJE, -OVANJE, -TOVANJE. Such an approach to the automatic generation of suffixes provided a total of 47,981 endings from the words in the file KNJIŽNICA, and a total of 38,846 suffixes from the words in the file POMLADNI.DAN.

The procedure of sorting these trial suffixes by decreasing frequency of occurrence, produced a total of 7,273 distinct endings for the file KNJIŽNICA, and a total of 5,330 distinct endings for the file POMLADNI.DAN. These results are also presented in the Table 4.11.

Text corpus	Suffixes		Distinct suffixes	
	abs	%	abs	%
KNJIŽNICA	47,981	100	7,273	15.2
POMLADNI.DAN	38,846	100	5,330	13.7

Table 4.11: A quantitative comparison of suffixes created from terms in databases POMLADNI.DAN and KNJIŽNICA

The *quantitative* comparison between these two lists shows very little difference between suffixes. Both lists of distinct endings account for a similar amount of suffix usage, i.e., 15.2% in KNJIŽNICA, and 13.7% in POMLADNI.DAN respectively. Moreover, a plot of the logarithm of the frequency of suffixes against their ranked order of frequency also yields a similar curve for both lists, as presented in Figures 4.4 and 4.5.

The shape of both curves is similar to diagrams produced for the frequency distribution of Slovene terms. This could again serve as an indication that a small number of endings accounts for a large amount of suffix usage; this point is very important for the design of the suffix list.

Apart from the fact that almost no quantitative difference was found between these two lists of suffixes, they were also characterized by many common *qualitative* features. For example, Table 4.12 displays the rank, frequency, and percentage of the 20

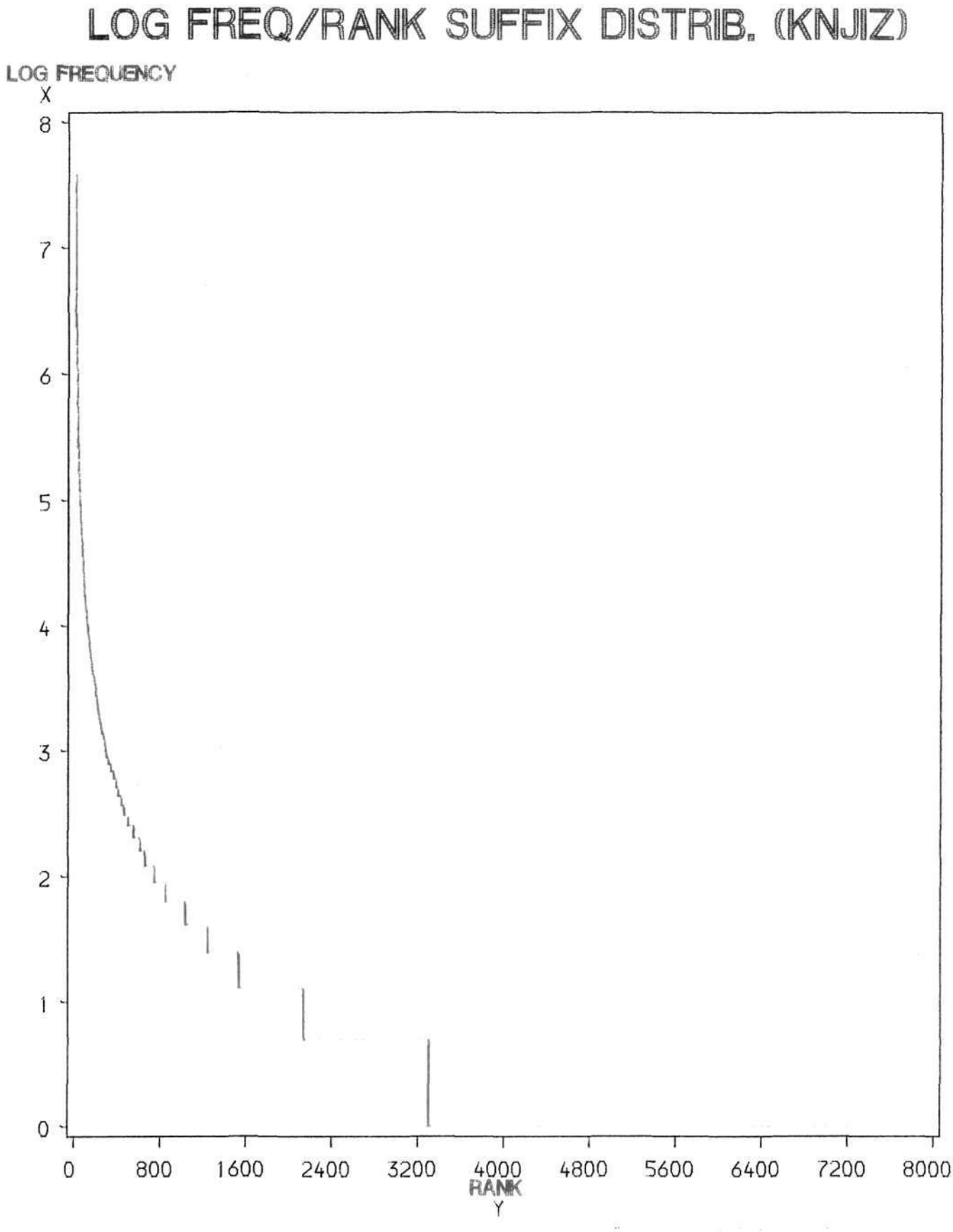

FIGURE 4.4
Plot of rank versus log of suffix frequency

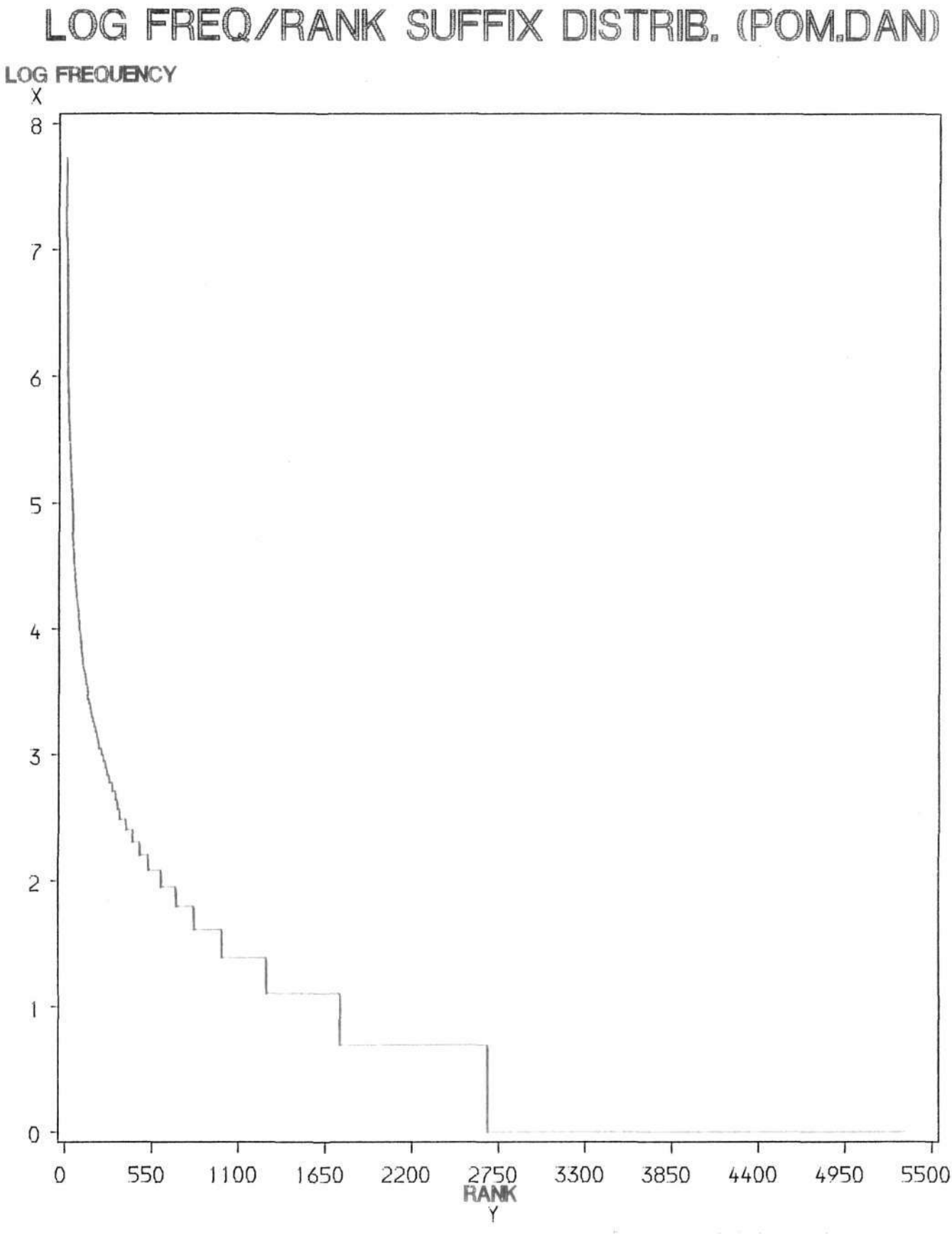

FIGURE 4.5
Plot of rank versus log of suffix frequency

most frequent endings in the file KNJIŽNICA and compares them with endings in the file POMLADNI.DAN. This table shows that most of the high-frequency endings in KNJIŽNICA also occur frequently at the top of the list of endings created from the file POMLADNI.DAN.

	KNJIŽNICA			POMLADNI.DAN		
Suffix	Rank	Freq	%	Rank	Freq	%
A	1	1978	4.1	1	2275	5.8
I	2	1953	4.1	2	1816	4.7
E	3	1711	3.6	4	1305	3.4
O	4	1511	3.1	3	1355	3.5
H	5	787	1.6	11	381	1.0
IH	6	668	1.4	19	265	0.7
M	7	593	1.2	7	660	1.7
JE	8	510	1.1	25	210	0.5
TI	9	474	1.0	12	381	1.0
JO	10	441	0.9	26	197	0.5
NE	11	435	0.9	22	239	0.6
NO	12	431	0.9	13	354	0.9
JA	13	424	0.9	31	173	0.4
NI	14	416	0.9	23	236	0.6
U	15	382	0.8	15	326	0.8
NA	16	371	0.8	17	280	0.7
V	17	337	0.7	43	111	0.3
GA	18	321	0.7	20	242	0.6
NIH	19	319	0.7	60	79	0.2
EM	20	305	0.6	18	272	0.7

Table 4.12: Frequency distribution of the 20 most frequent endings in KNJIŽNICA and their occurrence in POMLADNI.DAN

Some interesting points can be noted in Table 4.12, one of them being the fact that -A, -E, -I, and -O are the most frequent endings in both lists, accounting for 14.9% of suffix usage in the file KNJIŽNICA and, respectively, for 17.4% in the file POMLADNI.DAN.

Since there was no essential difference found between both lists of endings, a decision

was made to join together all distinct words from both text collections into one file and again generate a list of suffixes. This decision was based on the fact that both text collections had only 1,038 terms in common, and therefore, it was hoped that a larger vocabulary would produce a more useful list of endings.

Thus, all procedures for the automatic compilation of suffixes were repeated again, this time on the text corpus consisting of 19,888 distinct words; this corpus will be referred as SLOV. Initially, these procedures created a total number of 87,544 endings; when they were sorted into order and after elimination of duplicates, a list of 11,815 distinct suffixes was obtained.

A detailed analysis of the frequency distribution of suffixes provided some important results. Firstly, it was found that frequency of occurrence of trial suffixes declined very rapidly with increasing rank, as illustrated in Figure 4.6 which represents a plot of the frequency of individual suffixes in SLOV against their ranked order of frequency.

Secondly, a sampling from the trial suffix list showed that lower frequency trial suffixes were obviously not candidates for adoption in the suffix list. This point can be illustrated using two tables, Table 4.13 displaying the 20 most frequent endings, and Table 4.14 presenting 20 endings from the bottom of the trial suffix list.

While Table 4.13 displays a list of potentially useful endings—most of them are suffixes in the linguistic meaning—any employment of suffixes from Table 4.14 would by no means contribute to successful word conflation since they represent almost all characters from the given stem.

An inspection of the list of suffixes after they had been sorted into order of decreasing frequency of occurrence showed that the most frequent endings did, in fact, correspond to well-known suffixes. This suggests the use of a simple context-free, stemming algorithm which is also known as a *frequency algorithm.*

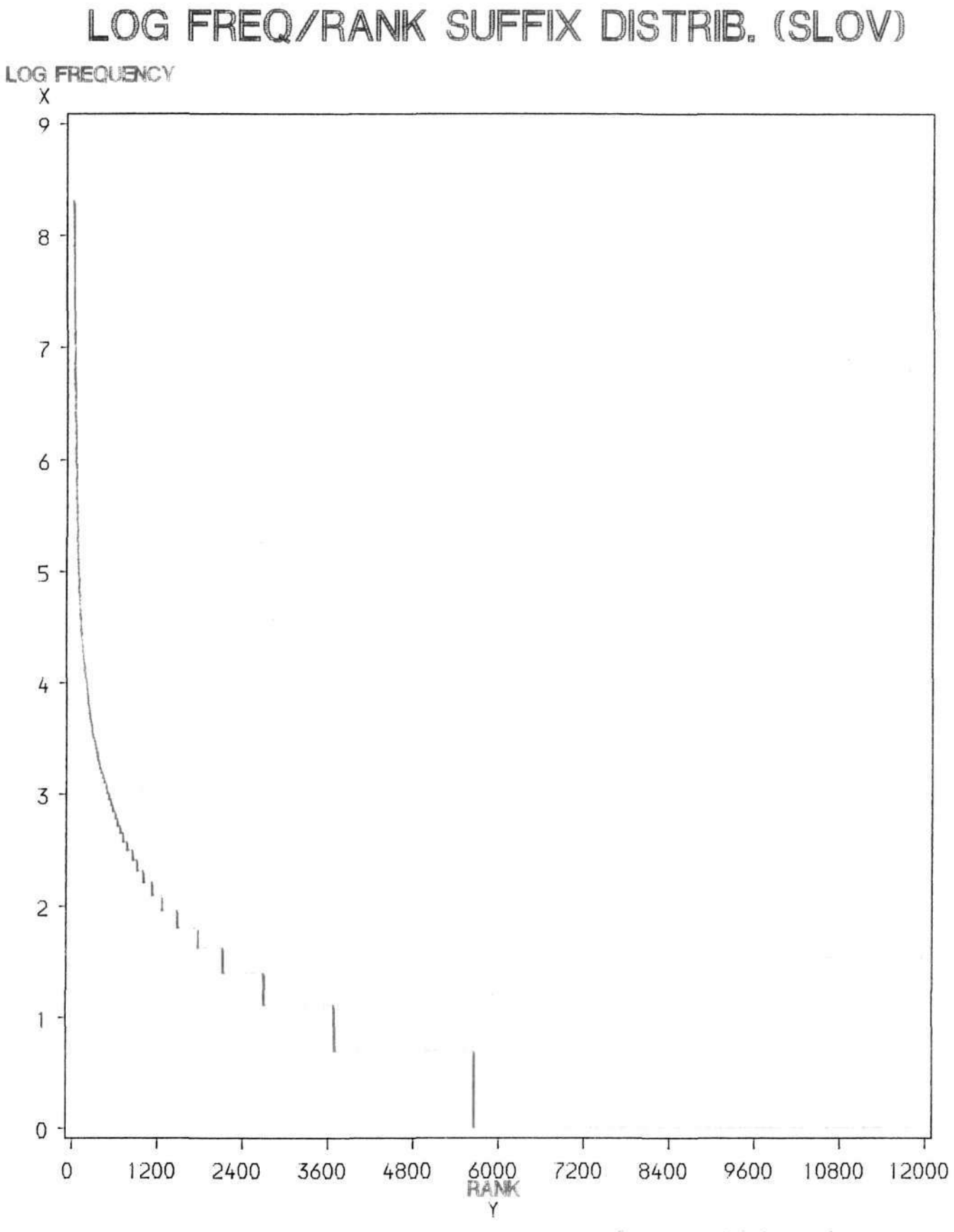

FIGURE 4.6
Plot of rank versus log of suffix frequency

Rank	Suffix	Freq	%
1	A	4052	4.6
2	I	3531	4.0
3	E	2879	3.3
4	O	2699	3.1
5	M	1220	1.4
6	L	1139	1.3
7	H	1121	1.3
8	LA	944	1.1
9	IH	900	1.0
10	TI	790	0.9
11	NO	740	0.8
12	JE	695	0.8
13	U	681	0.8
14	NE	646	0.7
15	NA	632	0.7
16	NI	626	0.7
17	LI	624	0.7
18	JO	609	0.7
19	JA	564	0.6
20	EM	561	0.6

Table 4.13: A list of the 20 most frequently occurring endings created from words in the file SLOV

4.4.2 Design of the frequency algorithm

Since no recoding or context-sensitive rules are required for the *frequency algorithm*, it was quite easy to design such a conflation procedure and use it on the words from the Slovene texts. The only restriction which was placed on the suffix removal was that the remaining stem should be of a minimum length of three characters. This approach is clearly crude in concept, as discussed by Lennon *et al.* (1981), but avoids the need for the detailed manual processing that characterizes most other ways of creating lists of suffixes.

The context-free algorithm operates as follows:

1. The suffix list is stored in reversed form and in alphabetical order.

Rank	Suffix	Freq	%
11631	ZRAVNAL	1	0.00114
11632	ZRAVNALA	1	0.00114
11633	ZREDNA	1	0.00114
11634	ZRL	1	0.00114
11635	ZRLA	1	0.00114
11636	ZTI	1	0.00114
11637	ZTRGAL	1	0.00114
11638	ZTRGALA	1	0.00114
11639	ZTRGALO	1	0.00114
11640	ZUME	1	0.00114
11641	ZUMLJIV	1	0.00114
11642	ZUMLJIVO	1	0.00114
11643	ZVAJALCEV	1	0.00114
11644	ZVALA	1	0.00114
11645	ZVALI	1	0.00114
11646	ZVEDB	1	0.00114
11647	ZVEDBE	1	0.00114
11648	ZVENEL	1	0.00114
11649	ZVENIJO	1	0.00114
11650	ZVIJA	1	0.00114

Table 4.14: An excerpt from the list of low frequency endings created from words in the file SLOV

2. The word to be conflated is read in, reversed, and the number of characters in the word is determined.

3. The last letter (first letter when reversed) of the word is noted and used to address that portion of the suffix list that contains suffixes commencing with this letter.

4. This portion of the suffix list is scanned to identify the largest suffix within it that matches the query word.

5. When a suffix is found that matches the end of the word, the stem length that would be left on removal of this suffix is examined.

6. If this length is less than the required minimum, i.e., three characters, the length of the suffix is reduced by eliminating its first letter (last letter when reversed),

and the suffix list is searched again for this smaller suffix;

7. If a suffix is found satisfying the minimum stem condition, then the suffix is removed, the resulting stem re-reversed and output; otherwise, no action is taken and the original word is output.

Evaluation of the frequency algorithm

Six sets of suffixes were generated, containing 100, 500, 1,000, 1,500, 2,000 and 3,000 suffixes, and then tested on 220 variants of the eight different word stems listed in Table 4.15. It has to be emphasized in this context that the term "stem" is not defined in the pure linguistic sense, but is specified as a string of characters to which variants of the basic stem can be reduced without altering its meaning. Thus, for example, a stem of variants FINANCE and FINANČNA is FINAN- since letters following the final -N differ from each other. Similarly, variants RAZVITOST and RAZVOJ have in common first four strings, i.e., RAZV-, which can consequently be defined as a stem.

The figures in this table show that, in general, the performance of the algorithm is far from satisfactory. The best overall results were obtained with the list containing 2,000 suffixes; even here, however, less than 40% of the words were conflated to the correct root. Particularly poor results were evident with roots having large suffixes, e.g., KNJIŽ, which has variants such as KNJIŽNIČARSTVO, KNJIŽNIČARSKEGA, etc. In addition, there was no consistent relationship between the size of the suffix set and performance.

An inspection of the stems resulting from the algorithm shows that both understemming and overstemming have occurred. For example, when the 2,000-suffix set (which gave the best overall results) is applied to the 45 variants of the root RAZV, the algorithm produces not only RAZV but also RAZ, RAZVI, RAZVIJ and RAZVO. The last three stems are all examples of understemming while the first is an example of overstemming, since RAZ is the beginning not only of RAZVOJ (DEVELOPMENT) but also of RAZLIKA (DIFFERENCE), RAZLOG (REASON) and RAZRED (CLASS), *inter alia*. The poor level of performance that is evident from Table 4.15 meant that

Stem	Number Of Variants	Number Of Suffixes					
		100	500	1,000	1,500	2,000	3,000
FINAN	19	2	7	10	10	9	7
KNJIŽ	37	2	5	7	7	9	12
KNJIG	14	5	4	3	5	5	5
RAZISK	30	0	10	12	11	10	8
RAZV	43	6	8	9	13	13	15
SPECIAL	26	7	2	5	4	7	5
SPECIF	9	1	6	7	8	9	9
UPORAB	42	11	19	19	20	21	18
Total	220	34	61	72	78	83	79

Table 4.15: Performance of the context-free stemming algorithm using different numbers of suffixes. The entries in the table give the number of variants conflated to the correct stem (as denoted by the string in the left-hand column of the table.)

the frequency algorithm is unlikely to achieve good results, in particular there is the problem of deciding upon a threshold for suffix selection. Since the employment of the frequency algorithm on the English-language-based bodies of text produced much better results than those described above (see, for example, Tarry, 1978; Lennon *et al.*, 1981), an experiment was carried out to find the main quantitative differences between English and Slovene endings.

A quantitative comparison between English and Slovene suffixes

An experiment to find out the main quantitative characteristics of the English and Slovene endings was carried out using library test collections, i.e., words from the files KNJIŽNICA and ENG.TEXT. Table 4.16 displays the different results that were obtained from the English and the Slovene test collections.

As can be seen from Table 4.16, the Slovene body of text, owing to the complexity of

Quantitative characteristics	KNJIŽNICA	ENG.TEXT
Total number of terms	59,088	55,460
Number of distinct terms	11,525	3,868
Number of terms after removal of stop-words	10,711	3,625
Total of generated endings	47,981	14,607
Number of distinct endings	7,273	2,437

Table 4.16: Quantitative comparison of the Slovene and English texts during the process of the automatic generation of suffixes

the Slovene morphology, produced a large number of word variants, and consequently a large number of distinct endings, when compared to the English text corpus. This difference makes the frequency algorithm much more suitable to the English language, particularly in the process of deciding upon a threshold for suffix selection.

A detailed insight into both lists of suffixes provides an additional argument for the impractical use of the frequency algorithm in Slovene retrieval systems. Table 4.17 illustrates the frequency of occurrence of automatically generated suffixes from the English and Slovene test collections according to the length of endings.

Although the increased length of suffixes correlated in both test collections with the decreasing percentage of suffix usage, as evident from Table 4.17, the following important differences between Slovene and English suffixes can be noted. While in the English test collection endings having up to three characters accounting for 67.9% of the total suffix usage, Slovene suffixes with the same length account for only 63.2% of the total suffix usage. Also while the Slovene endings having four, five, six, or seven characters account for 33.4% of the total number of suffixes, the English suffixes with this length account for only 29.6% of suffix usage. An important conclusion can be drawn from the above results. Since the Slovene language is characterized by a wider range of suffixes having a length of four letters or more than for English, it is extremely

Number of letters in the suffix	KNJIŽNICA		ENG.TEXT	
	abs	%	abs	%
1	10,683	22.3	3,602	24.7
2	10,393	21.7	3,410	23.4
3	9,220	19.2	2,899	19.8
4	7,017	14.6	2,023	13.8
5	4,681	9.8	1,249	8.6
6	2,805	5.8	702	4.8
7	1,560	3.2	357	2.4
8	765	1.6	181	1.2
9	393	0.8	96	0.7
10 +	464	1.0	88	0.6
Total	47,981	100.0	14,607	100.0

Table 4.17: Comparison of the frequency of suffix occurrence in KNJIŽNICA and ENG.TEXT according to the suffix length

difficult to define an appropriate threshold for automatic suffix selection.

Thus, any decision about the potentially useful set of endings to be employed in the frequency algorithm is bound, on the one hand, by a need to introduce large suffixes (i.e., suffixes exceeding the length of three characters), and on the other hand by the constant risk of overstemming. For example, adjacent words in the vocabulary, such as SLOJEVIT and BOJEVIT produce a highly desirable suffix -EVIT. However, one of its constituents substrings is also -VIT whose use is fatal for the term RAZVIT.

Having obtained unsatisfactory results for the employment of the frequency algorithm on the Slovene natural language text and defining some reasons for the failure of the algorithm, there was a need to introduce the second stage in the design process. This stage was primarily based on additional *manual* preprocessing, both in the con-

struction of the final suffix list, and in the formulation of context-sensitive and recoding rules.

Although a need for the manual involvement was anticipated on the basis of a discussion about the morphological structure of the Slovene language, as presented in Chapter 3, the low level of performance of the frequency algorithm necessitated a rethinking in the design of the new Slovene stemming algorithm. The only benefit achieved from the development of the frequency algorithm was a list of the most frequent endings which served as a starting-point for the design of the final suffix list and for the specification of the context-sensitive and recoding rules.

4.5 Design of the new stemming algorithm for the Slovene language

4.5.1 Development of the suffix list

The second algorithm was developed using the traditional, trial-and-error approach that characterizes most context-dependent algorithms. The process started by taking the 200 most frequent word endings that had been identified in the previous work and determining what extensions and rules needed to be created to allow them to stem correctly the 10,711 content-bearing words from the KNJIŽNICA corpus. The utility of each of these endings as a suffix was tested by seeing whether its removal would result in either understemming or overstemming. Consideration was given as to the minimum stem length which should be left after the removal of a given ending, of new endings which needed to be added to the suffix list or endings which needed to be removed from it, and of the context-sensitive and recoding rules needed for accurate conflation. It was often the case that selection of one suffix would require the adoption or removal of other suffixes, or the addition of context-sensitive rules in order to maintain consistency; this behaviour is, of course, characteristic of all languages and not specific to Slovene.

The following major problems were encountered:

- Selection of a minimum stem length. Most of the shorter words in Slovene preserve their meaning when only three characters remain (consider, for example, BAZ-a, RUS-ija, etc). Some terms, however, in particular those characterized by different types of morphemic alternation, require a minimum stem length of four characters (consider, for example, variants KADER and KADRA; their reduction to the stem KAD- would cause serious overstemming, i.e., KAD- could convey the meaning of both STAFF and TUB).

- It is sometimes necessary to take account of the characters immediately preceding an ending when deciding whether it should be removed from a word. For example, the identification of the very common suffix -BA should normally result in its removal, e.g., TELOVADBA, ODREDBA, POŠKODBA, IZOBRAZBA, etc. However, it should not be deleted when preceded by the character V to avoid overstemming: otherwise, e.g., the word STAVBA would be stemmed to STAV, which is the beginning of a large number of words, e.g., STAVA, STAVEC, STAVEK, STAVKA, etc.

- The wide range of morphemic alternations that occur during word formation and inflection in the Slovene language requires the use of very extensive recoding rules, much more so than is the case in English. Consider, for example, the following pairs of related words: OBSEG and OBSEŽEN, PREDLOG and PREDLAGATI, NAGRAJEVATI and NAGRADA, PODREJEN and PODREDITI, ODGOVARJATI and ODGOVOR, TEHNIČNI and TEHNIK, IZOBRAZBA and IZOBRAŽEVANJE, DOKAZ and DOKAŽE, ARKTIČNI and ARKTIKA, INTUICIJA and INTUITIVNOST, REGIJA and REGIONALNI, CITIRATI and CITAT, JETRA and JETER, TRG and TRŽEN, GESEL and GESLO, etc. Such examples are not amenable to conflation just by the deletion of the word ending; instead, a complex set of recoding rules is required.

As a result of the manual selection of suffixes, after consideration of the conditions for suffix removal, a list containing 2,086 endings was produced. An inspection of these suffixes showed that some of their inflectional variants were still missing. Thus, in order

to obtain a reliable and comprehensive final suffix list, all possible variants of each suffix were additionally produced, using declension, conjugation, etc.

The resulting longest-match, context-sensitive algorithm is based on the use of 5,276 endings, each of which has an associated minimum stem length, either three or four characters, and one of eight action codes, which implement the context-sensitive rules. The final list of suffixes can be also found in a machine-readable form on a floppy disk. While the list of stop-words is contained in the file STOP.TXT, a list of endings is included in the file SUFFIX.TXT. Table 4.18 shows an excerpt from the list of endings.

Suffix	Action code	*MIN* stem length
AVAM	3	4
AVAMA	3	4
AVAMI	3	4
AVANJ	3	3
AVANJA	3	3
AVANJE	3	3
AVANJEM	3	3
AVANJEMA	3	3
AVANJI	3	3
AVANJIH	3	3
AVANJU	3	3
AVATI	3	3
AVCA	2	4
AVCE	2	4
AVCEM	2	4
AVCEMA	2	4
AVCEV	2	4
AVCI	2	4
AVCIH	2	4
AVCU	2	4

Table 4.18: An excerpt from the final list of suffixes

As can be seen from Table 4.18, two digits are appended to each suffix. The first digit is called an *action code* which specifies conditions for the suffix removal; the second digit defines the *minimum stem length*, i.e., prevents the removal of the suffix if the resulting stem were less than the specified minimum length. In some cases the

minimum stem length is three and in other cases four characters. The employment of this list of suffixes by the new algorithm and explanation of some of the recoding rules are described in the section below.

4.5.2 The new stemming algorithm for the Slovene language

The stemming algorithm consists of two main parts: a basic stemming procedure and the recoding procedure. Both procedures are employed within the algorithm as follows.

The *basic stemming procedure* consists of a single-pass suffix deletion process; that is, one pass is made through a list of suffixes, and if a match is encountered, a deletion is made. The comparison proceeds in a longest-first sequence, to avoid incomplete truncation of compound suffixes. For example, as -SKEGA and -EGA are both in the suffix list, then the comparison should not be made with -EGA first, since -SKEGA would then never be detected.

Suffixes that are to be deleted appear in the suffix list, containing 5,276 endings. Appended to each suffix are an action code and a minimum stem condition. A list of suffixes in Table 4.19 illustrates all potential occurrences of both codes. These examples can serve as a basis for the explanation of the basic stemming procedure in the algorithm.

The first context-sensitive rule in the basic stemming procedure relates to the minimum stem length which should be left after the removal of the suffix. The minimum stem length can consist either of three or four characters. Although the majority of suffixes have the appended condition that the remaining stem should be of a minimum length of three characters, the morphological characteristics of some words required the minimum stem length to be extended to four characters and that condition is added to a certain number of suffixes. For example, a deletion of the suffix -ETI from the word ŽIVETI results in a stem ŽIV-; to prevent a potential overstemming if the suffix -AL were stripped from the word ŽIVAL, a code for the minimum stem length of four characters was attached to the suffix -AL to produce the unchanged stem ŽIVAL.

When a condition for the minimum stem length is satisfied in the algorithm, then

Suffix	Action code	*MIN* stem length
AL	3	4
ALNA	2	3
ATA	3	4
BA	5	3
EK	2	4
EM	8	4
ENE	6	3
ETI	3	3
IZACIJA	4	3
KACIJA	1	3
NA	2	4
OV	7	3

Table 4.19: Examples of suffixes and their digits codes

one of eight courses of action is followed, as determined by the action code associated with each suffix. These codes are as follows, with examples of their application in brackets:

1. Delete the terminal character of the word being processed which matches the suffix list entry (suffix -KACIJA: KLASIFIKACIJA to KLASIFI);

2. Delete as above if and only if the character preceding the matching entry in the word is a consonant (suffix -ALNA: NACIONALNA to NACION but SOCIALNA to SOCIAL);

3. Delete as in code 2, but not if there are two successive consonants preceding the matching entry (suffix -ATA: KANDIDATA to KANDID but KOLOVRATA to KOLOVRAT);

4. Delete as in code 2, but not if the consonant preceding the matching entry is -R (suffixes -NA, -ACIJA: POLARNA to POLAR but POLARIZACIJA to POLARIZ);

5. Delete as in code 2, but not if the consonant preceding the matching entry is -V

(suffixes -EK, -BA: STAVEK to STAV but STAVBA to STAVB);

6. Delete as in code 2, but not if the consonant preceding the matching entry is -M and is the third letter in the word (suffix -EME: VODENE to VOD but ZAMENE to ZAMEN);

7. Delete as in code 2, but not if the letters preceding the matching entry are either -SL, -BN, or -SN (suffix -OV: STANDARDOV to STANDARD but PRISLOV, OBNOV and OSNOV remain unchanged);

8. Delete as in code 2, but not if the letters preceding the matching entry are either -BL or -ST (suffix -EM: HITREM to HITR but PROBLEM and SISTEM remain unchanged).

Once an ending has been removed, the *recoding* procedure takes place. This consists of the following three steps, which are carried out in sequence:

1. A stem dictionary of six special cases is checked to see whether any of the transformations shown in Table 4.20 should be applied.

Stem	Recoded Stem
TIS	TISK
TRŽ	TRG
RAZVIJ, RAZVIL, RAZVIT	RAZVOJ
VZGAJ	VZGOJ
KEMIČ, KEMIK	KEMIJ
LOGIK	LOGIČ

Table 4.20: Stem dictionary of special cases.

2. A total of 20 recoding rules are applied, as shown in Table 4.21.

3. The $E \sim zero$ alternation, which has been mentioned in Section 2, is attended to, as shown in Table 4.22. This table considers only the $E \sim zero$ alternation and

Recoding Rule		Example	
Ending	Recoded Ending	Stem	Recoded Stem
-SEŽ, -SEČ	-SEG	PRESEŽ, PRESEČ	PRESEG
-LAG, -LOG	-LOŽ	PREDLAG, PREDLOG	PREDLOŽ
-GRAJ	-GRAD	NAGRAJ	NAGRAD
-REJ	-RED	PRIREJ	PRIRED
-GOVAR	-GOVOR	ZAGOVAR	ZAGOVOR
-NAŠ, -NOS	-NES	VNAŠ, VNOS	VNES
-NIŠ, -NIČ	-NIK	TEHNIŠ, TEHNIČ	TEHNIK
-IŠ	-IS	IZKORIŠ	IZKORIS
-BRAŽ	-BRAZ	IZOBRAŽ	IZOBRAZ
-KAŽ	-KAZ	DOKAŽ	DOKAZ
- TIČ	-TIK	OPTIČ	OPTIK
-UIT	-UIC	INTUIT	INTUIC
-ION	-IJ	REGION	REGIJ
-ČAN	-ČIN	OBČAN	OBČIN
-NAC	-NIR	SANAC	SANIR
-UŠ	-US	POSKUŠ	POSKUS
-VIR	-VOR	IZVIR	IZVOR
-STAJ, -STAL	-STAN	OBSTAJ, OBSTAL	OBSTAN
-STAT, -STOJ	-STAN	OBSTAT, OBSTOJ	OBSTAN
-SAB	-SOB	USPOSAB	USPOSOB
-TIR	-TAT	CITIR	CITAT

Table 4.21: Recoding rules for 20 word endings after removal of the initial suffix.

takes no account of the $A \sim zero$, $I \sim zero$ and $O \sim zero$ alternations that also occur; however, these are encountered much less frequently and some of the more commonly occurring alternations here are encompassed by the recoding rules of Table 4.21.

Having described the main components of the algorithm, the actual implementation is as follows:

1. The suffix list is stored as reversed suffixes, in alphabetical order; stems and suffixes from the recoding list are also reversed.

2. The maximum suffix length is noted (call this MAX).

Recoding Rule		Example	
Ending	Recoded Ending	Stem	Recoded Stem
-consonant + -R	-consonant + -ER	KADR	KADER
-consonant + -N	-consonant + -EN	JAVN	JAVEN
-consonant + -L	-consonant + -EL	GESL	GESEL
-consonant + -M	-consonant + -EM	POJM	POJEM

Table 4.22: Recoding rules for use with the four sonorants.

3. The word to be conflated is read and the number of characters counted; if the length of the word is less than three characters, it is output without any change, otherwise it is reversed.

4. The last MAX characters of the word (first MAX when reversed) are taken as a potential suffix.

5. The current potential suffix is searched for in the list of suffixes.

6. This search can have one of three possible outcomes:

   - The suffix is found. In this case, the minimum stem length condition and the action code are examined. If these are satisfied, then the ending is removed from the word. If they are not satisfied, then two courses of action are possible, these depending on the length of the remaining ending. If this is greater than one character, the first character (last when reversed) is eliminated and a new ending created; the algorithm is re-entered at Stage 5; alternatively, if the ending is now one character long, then the word to be stemmed remains unchanged and is sent to the recoding part of the algorithm.
   - The suffix is not found and it is more than one character long. In this case, the first character (last when reversed) is eliminated and a new suffix

created; the algorithm is re-entered at Stage 5.

- The suffix is not found and it is one character long. In this case, the word to be stemmed remains unchanged.

7. The three main parts of the recoding procedure are carried out in sequence on the stem, as detailed in Tables 4.20, 4.21 and 4.22

8. The stem resulting from the above transformations is re-reversed and output.

Evaluation of the stemming algorithm

To obtain a preliminary indication of how successful the design of the algorithm was and whether any major changes in the algorithm were necessary before its incorporation into INSTRUCT, a simple test was first carried out. The percentage of compression and the quality of the algorithm were measured and tested by applying the algorithm to 83 abstracts from the collection of documents that is fully described in Chapter 6. In order to acquire more reliable results, the idea was therefore to use the larger number of word types than those contained in 10 previously used abstracts from the KNJIŽNICA corpus. The results of this preliminary evaluation are discussed in the section below.

The level of compression achieved by the algorithm. After stop-words had been deleted from this collection, a total of 2,616 distinct word types was obtained. The employment of the stemming algorithm reduced the number of these words to 1,184 distinct stems, or 45.3%. If the level of compression is expressed in terms of the number of reduced words, then 54.7% compression was achieved. This figure is much higher than 12% as reported by Dimec (1988). It is also higher when compared to the results of different English stemmers where the level of compression ranged from 26.2% to 50.5% (Lennon *et al.*, 1981). There is no doubt that the Slovene stemming algorithm contributes significantly to the reduction of the body of text. These results also indicate that the algorithm is based on the use of a "strong" stemmer. Since "strong" stemming can hurt performance (Harman, 1991) it was interesting to observe how successfully

the stemming algorithm performed in terms of its ability to reduce word variants to the common stem; this performance of the Slovene stemmer is discused below.

Results of word conflation achieved by the Slovene stemmer. A list of 2,616 distinct word types and the resulting 1,184 distinct stems was given to a *trained intermediary* for the additional assessments. In other words, the professional intermediary was asked to control manually the results of the automatic word conflation. His main task was in discovering two main types of errors, i.e., under- and overstemming.

A total of 109 errors was reported by the professional intermediary. The success rate of suffix stripping was therefore 90.8%. These results are in accordance with Porter (1980) who emphasized that success rate is always significantly less than 100%, irrespective of how the process is evaluated. In order to test this statement, the English language-based equivalents to the above 83 abstracts were processed by Porter's algorithm. It is interesting to note that after English stop-words have been deleted from this document collection, a total of only 1,250 distinct word types was obtained. This again indicates the richness of the Slovene morphology; this difference is further discussed in Chapters 6, 7, and 8. In addition, the employment of the stemming algorithm reduced the total number of 1,250 word types to 1,065 distinct stems, or 85.2%. In other words, only 14.8% level of compression was achieved by Porter's algorithm. However, the results of experiments described in the next three chapters confirmed the assumption that the larger the dictionary size the greater the compression achieved. After Porter's algorithm has been applied to 4,756 word types within the INSTRUCT database, 36.7% level of compression was produced.

More importantly, the additional manual assessments of the resulting stems reported a total of 93 errors, leading to the 8.7% error rate. A summary of the results produced by the application of the Slovene (referred to as SLOV) and Porter's stemming algorithm (referred to as ENGL) is given in Table 4.23.

It is evident from Table 4.23 that both stemming algorithms were able—despite the huge difference in terms of the text compression—to produce a similar success rate

Text	Distinct terms	Distinct stems	Compression (%)	Error rate (%)
SLOV	2,616	1,184	54.7	9.2
ENGL	1,250	1,065	14.8	8.7

Table 4.23: Comparison of the Slovene and Porter's algorithms

for suffix stripping. While the Slovene stemming algorithm achieved a 90.8% success rate, its English counterpart's success rate was only slightly better, i.e., 91.3%. These findings correspond to the results of experiments carried out by Lennon *et al.* (1981). This research group demonstrated that there was no relationship between the strength of an algorithm and the consequent retrieval effectiveness (see also Keen, 1991b).

The results of the above experiment revealed that both stemming algorithms were vulnerable to a certain amount of errors produced during word conflation. The following are some examples of the unsuccessful word conflation achieved by the Slovene stemming algorithm (errors produced by the application of Porter's algorithm will be presented in Chapter 8):

- *overstemming*: BESED – besed, besedila; CENT – center, centralizirana; GLAV – glavo, glavni; GRAD – gradivo, gradnja, grajski; LIK – lik, likovni; NEM – nem, nemški; ODGOVOR – odgovoriti, odgovorna; OSEB – osebne, osebnosti; PODOB – podoba, podobni; POT – poteza, poti; PRAV – pravila, pravica; PROS – prosti, prostor; ROM – romih, romane; UMET – umetne, umetnost;

- *understemming*: avtomat – avtomatiz; dopol – dopoln; infor – inform; integ – integrir; instit – institut; jas – jasen; natan – natanč; poudar – poud; prim – primer; prog – program; razvitem – razvoj; tisk – tiskov.

Despite the 9.2% error rate, the results of this simple test have indicated that procedures used in the stemming algorithm are workable and will yield good results with

only minor changes. Although these alterations might involve the list of endings and occasionally the context-sensitive and recoding rules, the basic principles of the new Slovene stemming algorithm remain the same. Good performance results are demonstrated below.

For example, the abstract listed previously resulted in the following stemmed text representative:

UPORAB ONLIN JAVEN DOSTOP KATAL
JOŽE KOKOLE
PREDSTAV FENOM ONLIN JAVEN DOSTOP KATAL KRAT OPAC ANGL ONLIN PUBLIC ACCES CATALOGUE RAČUNAL PODPR POS KNJIŽ KNJIŽ SISTEM NASTAN RAZVOJ STAN RAZVOJ SREDIN NAČEL KARAKTER POJAV OBLIK OBDEL UPORAB OPAC GENER KNJIŽ VZAJEM KATAL ODNES ONLIN BIBLIOGRAF SERVIS PROBLEM KON UPORAB UPORAB OPAC ZAHTEV POGOJ OBLIK UČIN UPORAB PRIJAZ ISKAL DIAL PERSPEK UVAJ OPAC KATAL

Apart from producing useful stems—as shown above—the algorithm has also nicely captured the characteristics of the Slovene morphology. In other words, the algorithm has demonstrated that it can conflate most of the "difficult" word variants to the same stem. Table 4.24 illustrates the performance of the algorithm, using some examples. The listing consists of four basic stems, having in total 29 different variants as they appeared in the document collection. All words are organized in three columns. The first column shows the original word, the second column the initial stem produced after the basic stemming procedure, and the third column the recoded stem, i.e., the final output.

Each of the examples in Table 4.24 illustrates the employment of both the basic stemming procedure and recoding rules. For example, stems CITAT- and REGIJ- illustrate alterations occurring at the end of the stems after the use of recoding suffixes; stem GESEL- describes the insertion of -E between the consonant and -L at the end of the stem; a stem KEMIJ- represents a type of correction which is carried out when a list of stems to be changed is employed.

Term	Initial stem	Final stem
CITATIH	CITAT	CITAT
CITATOV	CITAT	CITAT
CITIRAMO	CITIR	CITAT
CITIRANA	CITIR	CITAT
CITIRANEGA	CITIR	CITAT
CITIRANEM	CITIR	CITAT
CITIRANI	CITIR	CITAT
CITIRANIH	CITIR	CITAT
REGIJA	REGIJ	REGIJ
REGIJI	REGIJ	REGIJ
REGIONALNE	REGION	REGIJ
REGIONALNE	REGION	REGIJ
REGIONALNI	REGION	REGIJ
REGIONALNIH	REGION	REGIJ
REGIONALNO	REGION	REGIJ
GESEL	GESEL	GESEL
GESELSKI	GESEL	GESEL
GESLA	GESL	GESEL
GESLI	GESL	GESEL
GESLO	GESL	GESEL
GESLOM	GESL	GESEL
GESLU	GESL	GESEL
KEMIJA	KEMIJ	KEMIJ
KEMIJE	KEMIJ	KEMIJ
KEMIJSKA	KEMIJ	KEMIJ
KEMIJSKE	KEMIJ	KEMIJ
KEMIJSKO	KEMIJ	KEMIJ
KEMIKI	KEMIK	KEMIJ
KEMIČNIMI	KEMIČ	KEMIJ

Table 4.24: Recoding strategies and suffix deletion applied to some examples

Finally, the effectiveness is also demonstrated by reference to the eight sets of word variants that have been discussed previously in the evaluation of the earlier, context-free algorithm. When the context-sensitive algorithm described in this section was used, all of the 220 variants were correctly conflated to the stem that is shown in Table 4.15.

However, to obtain the final results about its retrieval performance the first objective was to incorporate the Slovene stemming algorithm into INSTRUCT. The effectiveness of the routines developed here will be evaluated by running searches on a Slovene database; these results will be then compared with results obtained by applying the manual right-hand truncation and non-conflation to words in queries.

Chapter 5

INSTRUCT: an INteractive System for Teaching Retrieval Using Computational Techniques

5.1 Introduction

Schools of librarianship and information studies have been an important source for the development of alternative searching techniques. As a part of information retrieval courses, many teaching aids have been designed in the past to simulate searching on databases (Wood, 1984). However, all simulation models have been developed to familiarize students with conventional, Boolean-based retrieval systems. Thus, the lack of a teaching aid which would help students to become familiar with advanced retrieval techniques has become more than evident.

It is precisely this fact which has led to the design of INSTRUCT (INteractive System for Teaching Retrieval Using Computational Techniques) at the Department of Information Studies, University of Sheffield. INSTRUCT is an interactive system

which has been developed mainly to enable students of librarianship and information studies to become familiar with the new generation of computerized, statistically-based, retrieval systems (Willett and Wood, 1989). INSTRUCT is now being used for this purpose in educational organizations both in the UK and abroad. In addition to its use as a teaching resource, INSTRUCT has also proven to be a very useful test bed for investigating a range of research problems encountered in information retrieval.

This chapter will be concerned mainly with a description of INSTRUCT. Firstly, a brief outline of the main components of the original version of INSTRUCT (Hendry *et al.*, 1986a,b) will be given, followed by an explanation of modules which were added to INSTRUCT (Wade and Willett, 1988). This will form a basis for the description of INSTRUCT both as a teaching resource and as a test bed for research problems. The chapter will be concluded by a brief summary of the main modifications to the original version of INSTRUCT which were needed to develop a new, Slovene version of INSTRUCT.

5.2 Original version of INSTRUCT – program facilities

The original version of INSTRUCT was implemented in the Summer of 1985 on a PRIME 750 minicomputer under the PRIMOS operating system. This version incorporates the following facilities: natural language query processing (including the elimination of stop-words, automatic word conflation and automatic identification of word variants), best-match searching, relevance feedback searching, and Boolean searching.

This version of INSTRUCT runs against a search file that comprises 6004 documents from the 1982 additions to the Library and Information Science Abstracts (LISA) database. Each of the records in this search file contains the accession number, the title and the abstract of a document; the shortage of disk space prevented the inclusion of the full citation data. The retrieval of these records is based on the occurrence of terms in titles and abstracts of documents. INSTRUCT uses the conventional inverted file structure.

5.2.1 The user interface

In the original, i.e., PRIME version of INSTRUCT, the following two types of user interface are available:

- the "novice" interface;
- the "experienced user" interface.

The "novice" interface is completely menu-driven and provides the user with a great deal of explanatory information at all stages during the search. The "experienced user" interface is also menu-driven, but presents the user only a limited amount of explanatory text.

5.2.2 Query formulation

The query module of INSTRUCT allows the user to input a natural language need statement as a basis for the query to be searched. Thus, no Boolean operators need to be specified (although this can be done later if a Boolean search, rather than a best-match search, is required). The key terms in the query list are identified after non-content bearing words (e.g., AND, BUT, ARE) have been eliminated. Those terms not found in the stop-word list are then stemmed using an algorithm suggested by Porter (1980). The resulting list of stems is then shown with the corresponding frequencies of occurrence in the 1982 LISA database. The user can change the query by selecting one of three options: the addition, deletion, or expansion of query terms. The addition or deletion of terms in the query list is achieved very simply by updating the data structure in which the current form of the query list is stored.

The third option, i.e., the *expansion* of a query term, is based on a measure of the similarity between strings of text. The assumption for this type of term expansion is that the character structure of the word is related to its meaning so that it can be used as a basis for classifying that word. The measure of similarity is based on the number of trigrams (i.e., three character substrings) common to the selected keyword and each of

the stems in the dictionary component of the inverted file. Similarity is calculated using the highly efficient best-match searching algorithm suggested by Noreault *et al.* (1977). The stems in the dictionary file are then sorted into order of decreasing similarity with the selected query stem. The ten most similar stems are then displayed to the user for possible inclusion in the query list.

5.2.3 Searching

Once a set of keyword stems has been obtained that adequately describes the query, the user is in a position to carry out a search of the database. There are two main types of search strategy implemented in the original version of INSTRUCT: the best-match search and the Boolean search.

Best-match searching

A nearest neighbour, or best-match, search procedure forms the basic searching mechanism available in INSTRUCT. The following is the summary of the main actions which take place in best-match searching: the presence or absence of each of the query terms in each of the documents in the database is noted (documents having no similarities with the query are eliminated), the sum of the weights for these terms is calculated, and then these sums of weights are sorted so as to obtain a ranking of documents.

The best-match searching procedure in INSTRUCT is based on the algorithm suggested by Noreault *et al.* (1977), and discussed further by Perry and Willett (1983). The weights used in INSTRUCT to reflect the discriminating power of individual query terms are the modified inverse document frequency (IDF) weights, first suggested by Croft and Harper (1979).

After performing an initial, best-match search, the INSTRUCT user has an opportunity to perform further searches based on modified term weights in an attempt to improve retrieval performance. This searching mechanism is described in the section below.

Relevance feedback searching

As each document is displayed on the screen in ranked order, the user is asked to state whether or not the retrieved document is relevant to the query. The relevance information provided by the user then forms the basis for the modification of the term weights in order to obtain a better ranking of the remaining, uninspected documents in the collection.

The relevance feedback technique used in INSTRUCT is based on the approach first suggested by Robertson and Sparck Jones (1976), which takes account of the presence or absence of query terms in both relevant and non-relevant documents.

Possible alternatives to the relevance feedback search are to go further down the initial ranked list, to modify a query by the addition, deletion, or expansion of terms, or to carry out a Boolean search as described in the next section.

Boolean searching

The conventional Boolean search based on the AND, OR, and NOT operators (but without proximity searching) has been included in INSTRUCT mainly to allow students to compare the Boolean retrieval mechanism with the best-match searching facilities.

However, INSTRUCT also allows the user to receive a ranked Boolean output, a facility that is available in some free-text retrieval packages (Kimberley, 1987). This is done by the user specifying Boolean constraints that must be satisfied by the output from the best-match search. The aim of this hybrid search is to exclude from the ranking those documents which do not fulfill the Boolean constraints.

5.3 Enhancements to INSTRUCT

This section will briefly describe the new version of INSTRUCT, which includes three main additional modules: query expansion based on co-occurrence data, cluster-based searching, and browsing. The implementation of these new components was in line

with the objective of providing students with live "hands-on" experience of searching using advanced methods in IR that are not otherwise generally available, i.e., that are described only in research articles and monographs.

This version of INSTRUCT has also been written in the standard PASCAL programming language; it runs on an IBM 3083 mainframe under the VM/CMS operating system. The search file contains the 26,280 records comprising the 1982-1985 input to the Library and Information Science Abstracts (LISA) database. As with the original version, the IBM version of INSTRUCT is based on a conventional inverted file structure.

5.3.1 The user interface

In addition to the "novice" and "experienced user" interfaces which are available in the original version of INSTRUCT, an "expert" command-driven interface has been incorporated into the IBM version of INSTRUCT. This interface—in which help information is available only as a specific option—has been introduced to reflect the needs of students and members of staff who are familiar with the techniques demonstrated in INSTRUCT. These users see the system as a rapid way of getting references, rather than as a way of learning about advanced retrieval research.

5.3.2 Query expansion on the basis of term co-occurrence

Besides the expansion module based on the string similarity measure (i.e., the number of trigrams in common), the IBM version of INSTRUCT contains an expansion routine which makes use of term clustering techniques. Here, the identification of the most similar stems to a selected query term is based on the extent to which a stem co-occurs with a query term throughout the database.

The algorithm which has been used for the calculation of the inter-keyword similarities is one proposed by Willett (1981); this is derived from an earlier algorithm for the calculation of query–document similarities which had been proposed by Noreault *et al.* (1977). The algorithm uses the inverted file to identify all of the documents in

which a given term occurs.

Since the database in INSTRUCT is static, the 20 nearest neighbours for each of the keyword stems were calculated in a single, batch run and then stored in a file to achieve quick response. Thus, after 20 terms which are judged to be most similar to the chosen query stem are displayed for inspection, the user can include any stem in the query list. The algorithm used here is efficient in operation since it removes the need to calculate similarities between pairs of stems which do not co-occur in any of the documents in a database. This algorithm is probably the most efficient currently available but would be very demanding of computational resources if used for the interactive identification of related stems, particularly in the case of very frequently occurring stems, where many document vectors need to be added together and where a disc access would probably be required for each and every occurrence of the keyword in the database (Willett, 1981).

A more efficient technique, derived from the algorithm used here, has been suggested by Noreault and Chatham (1982) which might be sufficiently fast for interactive use. However, the algorithm can only be used with low frequency keywords; moreover, the similarities are not completely accurate since they are calculated by means of a sampling procedure (Wade and Willett, 1988).

5.3.3 Cluster-based searching

A third search option available in the IBM version of INSTRUCT (besides the best-match and Boolean searching) involves the clustering of the documents in the database. This procedure uses the concept of a nearest neighbour cluster, or NNC, as discussed by Griffiths *et al.* (1984), which has been shown to be of general applicability.

5.3.4 Post-search options

The post-search mechanism in the IBM version of INSTRUCT contains the following modules:

- constraining the results of a best-match search or of a cluster search using Boolean logic (hybrid search);
- browsing;
- performing feedback searches.

Hybrid searching

In the IBM version of INSTRUCT, the *hybrid search* is carried out by eliminating from the initial ranking (achieved by a best-match search, or cluster search) any documents which fail to satisfy the Boolean constraints. These can be imposed and released any number of times after the initial ranking has been produced so that the user can experiment with a number of different sets of constraints.

Browsing

The importance of *browsing* in information retrieval systems has been widely recognized (Hildreth, 1982; Bawden, 1986). Browsing should allow the user to follow up a particularly interesting document without losing the main thread of the query. In particular, browsing exploits the tendency, inherent in retrieval systems, for the system to identify "fringe" material, which may be related to the user's query in unexpected ways (Wade and Willett, 1988). In the IBM version of INSTRUCT, the user can invoke a browsing option after identifying one or more relevant documents in the initial search. INSTRUCT then allows any of these documents to be used as the basis for either a chain search or a seed search.

Chain searching involves following the chains of related documents which thread their way through the file of nearest neighbours; this file was set up for the NNC searching routine. When a document is selected by the user as the basis for the browsing option, the nearest neighbour to this document is displayed, then that document's nearest neighbour and so on until the chain doubles back on itself, i.e., until a pair of reciprocal nearest neighbours is encountered (Murtagh, 1983).

Alternatively, *seed searching* results in a best-match search being executed in which the original query stems are replaced by the stems in the title and abstract of a document which has been chosen by the user, i.e., a relevant document identified in the initial search. This second search results in a ranking of documents in order of decreasing similarity with the seed; documents which have already been seen are then eliminated from the ranking. The remaining documents can be viewed by the user and the new documents can, in their turn, be selected for another seed search. To ensure a rapid response to a request, INSTRUCT selects only the 25 stems with the lowest postings (and hence the highest specificity) of the seed document.

Feedback searching

The relevance information obtained from the initial search and, possibly, from a browse search, is used by INSTRUCT to modify the original query in two ways: by modifying the weights associated with individual query stems (as described in previous sections), and by allowing the user to add to the query any stems which have tended to occur in documents they considered to be relevant.

In this second option, known also as a relevance-based query expansion (Wade and Willett, 1988), rather than calculating the relevance weights just for the original query stems, the weights are calculated for all of the stems which occur in any of the documents which have been identified as being relevant to the query. These stems are then sorted into order of decreasing weights and the twenty top-ranked stems displayed for the user to add to the original query as required.

5.4 Main modules of the INSTRUCT package

It can be seen from the above sections that both the original PRIME version and the IBM version of INSTRUCT allow the demonstration of many information retrieval techniques which have been studied intensively in information retrieval research over the last two decades. Since the IBM version of INSTRUCT includes all modules developed

for the original version (the only exception is a guided search), the main components of INSTRUCT can be illustrated with the following list (words in italics indicate facilities which were added to the original version):

- User interface:
  - novice
  - experienced user
  - guided search (not included in the IBM version)
  - *expert*
- Query formulation:
  - natural language input
  - exclusion of stop-words
  - automatic word conflation
  - assignment of initial weights to stems
- Query reformulation:
  - addition of query terms
  - deletion of query stems
  - expansion of query stems:
    - string similarity (trigrams)
    - *co-occurrence data*
- Searching:
  - best-match search
  - Boolean search
  - *cluster-based search*
- Post-search options:

- hybrid search (imposing Boolean constraints)
- *browsing*:
  - *chain search*
  - *seed search*
- feedback search:
  - reweighting query stems
  - *weighting stems in relevant documents* (*relevance-based query expansion*)

As has already been noted, both versions of INSTRUCT have been widely used as a teaching aid and as a useful test bed. The next sections will briefly describe how these functions of INSTRUCT have been used at the Department of Information Studies, University of Sheffield.

5.5 The use of INSTRUCT at the University of Sheffield

5.5.1 Use in teaching programmes

The main use of INSTRUCT in teaching programmes is in relation to the information storage and retrieval courses. Over the years, the Department has developed several different simulations for on-line searching (Wood, 1981). Simulations have been designed to familiarize students with information retrieval techniques that they are likely to encounter when they enter employment. Their great value is in providing stress-free, hands-on practice for large number of students before they use the real on-line service. However, all the simulations (the latest is DIAL-SOS which simulates DIALOG searching on IBM-compatible microcomputers) have been designed to demonstrate conventional, Boolean retrieval methods.

The encouraging results reported from experimental information retrieval research led to the idea of developing INSTRUCT, primarily as a teaching aid, in order to achieve the following aims (Willett and Wood, 1989):

- to allow students to use some of the more advanced retrieval techniques which were not widely used by the on-line services,
- to allow students to specify real queries that could be searched against a document database of non-trivial size.

Thus, the IBM version of INSTRUCT is introduced to the students in the first term with two main purposes:

- to become familiar with the routine usage of a computer-based information retrieval system. The main advantage of INSTRUCT lies not only in the possibility of searching documents in a subject area which is familiar to students, but also in allowing the students to search a large database with real queries at no cost;
- to become familiar with the differences between Boolean and best match searching. Although INSTRUCT was primarily designed to demonstrate the latter facilities, the Boolean module (based on AND, OR, and NOT operators) is quite sufficient to serve as a teaching aid to students. The resulting skills can then be used in real database searching (e.g., usage of the CD-ROM version of the LISA database).

In addition, INSTRUCT forms a primary teaching resource for illustrating advanced information retrieval techniques (e.g., index term weighting schemes, automatic relevance feedback, word stemming) in elective courses in the second term.

5.5.2 Use in research programmes

Since its initial implementation, INSTRUCT has also been shown to be a useful test bed for investigating research problems in information retrieval. The following is a list of some experiments which have been carried out recently:

- Comparisons of the effectiveness of Boolean and best-match searching; these comparisons have been done both qualitatively (Nelis, 1985) and quantitatively (Mohan, 1987).

- A comparative evaluation of the effectiveness of searching using a knowledge-based information retrieval system (i.e., PLEXUS, see Vickery *et al.*, 1987; in a later stage, its commercial implementation, i.e., Tome Searcher was also used) and a statistically-based system (i.e., INSTRUCT); the results of this experiment are described by Wade *et al.*, 1988.

- Best-match searching using text signatures (Wade *et al.*, 1989).

- Full text searching, providing the ranking of paragraphs within a document (Al-Hawamdeh and Willett, 1989).

5.6 Processing of documents and queries in a Slovene language free-text retrieval system

The processing routines in INSTRUCT are, in very large part, independent of the actual language in which the texts have been written. The only exception is the stemming algorithm which has to take account of the morphological structure of the particular language. The original version of INSTRUCT is thus based on the stemming algorithm developed for the English language by Porter (1980).

This feature of INSTRUCT—together with the fact that the package has been written in a standard PASCAL programming language—indicates that it should be relatively easy to convert INSTRUCT to allow best match searching of text in any particular language that uses suffixing to create word variants in a manner analogous to that of English. Thus, to implement a Slovene language-based free-text retrieval system as the main goal of this PhD thesis, two important stages of research work had to be carried out:

- design of a stop-word list and development of a stemming algorithm for the Slovene language;

- modification of the English-based version of INSTRUCT in order to incorporate the possibility of processing Slovene documents and queries.

Since the first stage of research work (i.e., stemming algorithm and stop-word list) is described in detail in Chapter 4 (see also Popovič and Willett, 1990), the next section will briefly outline the main modifications of the original version of INSTRUCT.

5.6.1 The Slovene version of INSTRUCT

The main idea behind the development of a stop-word list and a stemming algorithm for the Slovene language was to provide end-user access to bibliographic databases (written in Slovene) using best-match searching techniques. It was, therefore, assumed that retrieval modules implemented in the original (PRIME) version of INSTRUCT should be sufficient to incorporate the Slovene stemming algorithm, and also to demonstrate its performance effectiveness. In addition, a decision to implement the Slovene version of INSTRUCT on IBM PC-compatible hardware supported the idea of using the original, PRIME version of INSTRUCT, and not the later IBM version. The conversion of all modules available in the IBM mainframe version would result in serious PC memory allocation problems and thus lead to time-consuming re-writing of the whole program.

The Slovene version of INSTRUCT was thus implemented on the basis of the conversion of the original (PRIME) version of INSTRUCT to an IBM PC compatible microcomputer using the TURBO PASCAL 5.5 programming language. This means that the Slovene version of INSTRUCT consists of the following main modules:

- natural language query input (in Slovene);
- elimination of non-content bearing terms from the query (using a dictionary of 1,593 Slovene stop-words);
- stemming of remaining query terms (using the algorithm described in Chapter 4);
- morphological term expansion using a string similarity measure;
- best-match searching (with the possibility of imposing Boolean constraints after the initial search has been carried out);

- relevance feedback searching;
- Boolean searching.

The major alterations to the original version of INSTRUCT were carried out in the language-dependent modules in order to achieve the main goal, i.e., the successful processing of Slovene terms both in queries and in documents. Some other modifications which could reflect the substantial developments in information retrieval research since the design of the original version of INSTRUCT (e.g., development of a WIMP-based interface) were therefore not implemented. Any qualitative evaluation of the Slovene version of INSTRUCT should take into account these limitations.

Creation and processing of the Slovene document collection

It has already been noted that the document collection for the original version of INSTRUCT consisted of 6,004 records from the LISA database. Each record consisted of three main fields, i.e., the title, abstract and accession number. Since this document collection represents an example of a database written in the English language a primary task in developing a Slovene version of INSTRUCT was to find an adequate Slovene substitute for the LISA database.

Since no less than 27 bibliographic databases have been created over the last few years by specialized information centres and libraries in Slovenia, as reported by the Research Community of Slovenia (1989), it was expected that one of these databases could serve as a test collection for the design of the Slovene version of INSTRUCT. However, the absence of abstracts from these databases (i.e., documents are described by only basic bibliographic units as, for example, a title, keywords, etc.) could present serious limitations in evaluating the retrieval performance of both the Slovene stemming algorithm and best-match searching. In order to run INSTRUCT against a search file having a larger amount of free text (e.g., abstracts from articles) it was thus decided to build a new test document collection. This document collection consists mainly of articles from two journals, i.e., *Knjižnica* (217 articles, covering the period 1972-1990) and *Informatologia Yugoslavica* (287 articles, covering the period 1969-1989). It should

be noted that *Informatologia Yugoslavica* contains articles written in all Yugoslav languages, and consequently, some of the units had to be translated into Slovene before the actual input. The document collection covers the area of librarianship and information science and contains 504 units; each unit is represented by the identification number, title, source, and abstract.

The processing of this document collection was carried out in a manner similar to the processing of the LISA document collection, i.e., a series of programs were used to create an inverted file. Some of these programs needed alterations, particularly those related to the processing of the Slovene language (e.g., the program dealing with the exclusion of stop-words from documents and the program responsible for the stemming of the remaining terms in documents). In summary, an inverted file, consisting of a dictionary, posting and display files, also forms a basis of the Slovene version of INSTRUCT. This version was indexed by 2,957 stems.

Modifications to the source code of INSTRUCT

The original version of INSTRUCT was written in modular form using the standard PASCAL programming language to allow both portability and ease of any future modifications or enhancements. The only exceptions are the disk-handling routines, which are inevitably system-dependent.

It was thus quite easy to transfer the source code of INSTRUCT to an IBM PC compatible microcomputer (MS-DOS operating system) using the TURBO PASCAL 5.5 programming language. The following are some modifications needed to accommodate MS-DOS file handling routines and the TURBO PASCAL programming style:

- substitution of PRIMOS disk-handling routines with MS-DOS routines (e.g., routines for opening or closing files);
- division of the large source code of INSTRUCT into smaller units as required by TURBO PASCAL;
- application of STRING variables throughout the whole program.

However, some major modifications of the program were also required, particularly in order to incorporate facilities for handling a large stop-word list and, most importantly, automatic conflation of Slovene terms. As described in Chapter 4, the Slovene stemming algorithm is based on the longest-match principle using a list of 5,276 endings with associated context-sensitive rules. In addition, three types of recoding rules are applied after suffix deletion. This approach differs very much from Porter's stemming algorithm, which is based on the iteration principle. Thus, considerable alterations were carried out in the modules dealing with automatic word conflation. To summarize, two-months' work by the author was required for both building a document collection and producing a Slovene version of the INSTRUCT package.

Despite the fact that stemming of Slovene words could be potentially demanding of computational resources (since word conflation is carried out in two steps using a large dictionary of suffixes) the initial results are quite encouraging. It seems that both the design and the implementation of the Slovene stemming algorithm within the INSTRUCT package were able to achieve the most important goal, i.e., a balance between the quality of algorithm and simplicity and efficiency of processing.

In order to demonstrate both the performance of the stemming algorithm and some other retrieval facilities the next section illustrates the search process carried out using the Slovene version of INSTRUCT.

5.7 An example of best-match searching using the Slovene version of INSTRUCT

This section gives an example of the use of INSTRUCT to carry out a best-match search for the query VISOKOŠOLSKO IZOBRAŽEVANJE IN BIBLIOTEKARSKI KADRI (i.e., UNIVERSITY EDUCATION FOR LIBRARY STAFF). The example has been taken directly from the output displayed on the screen during the execution of the search. The output has been edited to eliminate many blank lines and the repeated display of the list of stems. User inputs are in italics (INSTRUCT commands are writ-

ten in English to allow the English reader to follow the execution of the search process; for the user in Slovenia they are, of course, written in Slovene).

> *INSTRUCT*

⋆ ⋆ ⋆ ⋆ WELCOME TO INSTRUCT ⋆ ⋆ ⋆ ⋆
(An INteractive System for Teaching Retrieval Using Computational Techniques)

If you

- are an experienced user of INSTRUCT, type : 1
- would like the opportunity to see some explanatory text at various points throughout the search, type : 2

Please type response now:
> *1*

Please input a query ending with a #:
> *Visokošolsko izobraževanje in bibliotekarski kadri #*

The current form of the search term list is
as follows:

1 VISOKOŠOL (18)	2 IZOBRAZ (73)
3 BIBLIOT (66)	4 KADER (24)

To:

- add any terms, type : 1
- delete any stems, type : 2
- let the system find some related terms, type : 3
- perform a BEST-MATCH search, type : 4
- perform a BOOLEAN search, type : 5
- leave INSTRUCT, type : 6

Please type response now
> *3*

You now have the opportunity to expand your query

- If you select one of the stems shown below then a group of 10 similarly spelt (or misspelt) terms from the index will be displayed to you.
- You can then add any useful stems to your list of search terms.

To expand on any term type its NUMBER and press RETURN
otherwise type # and press RETURN

> *2*

THE SEARCH IS NOW BEGINNING
it might take some time so please wait

You have chosen to search on IZOBRAZ
and the following stems have been retrieved:

1 PREDIZOBRAZ (1)	2 SAMOIZOBRAZ (1)
3 SPLOŠNOIZOBRAZ (36)	4 VISOKOIZOBRAZ (1)
5 OBRAZ (5)	6 PREOBRAZ (4)
7 IZRAZ (6)	8 FRAZ (1)
9 IZKAZ (1)	10 OBRAZLOŽ (1)

To include any of these new terms in your query type in their
numbers finishing with a # and press RETURN; e.g., 1 2 3 #.
Otherwise just type # and press RETURN

> *4#*

The search term list is now as follows:

1 VISOKOŠOL (18)	2 IZOBRAZ (73)
3 BIBLIOT (66)	4 KADER (24)
5 VISOKOIZOBRAZ (1)	

To expand on another term type in its NUMBER and press RETURN.
To leave this section(and to return to the main menu) type # and press RETURN.
> *#*

To:

- add any terms, type : 1
- delete any stems, type : 2
- let the system find some related terms, type : 3
- perform a BEST-MATCH search, type : 4
- perform a BOOLEAN search, type : 5
- leave INSTRUCT, type : 6

Please type response now
> *4*

BEST-MATCH SEARCH

THE SEARCH IS NOW BEGINNING
it might take some time so please wait

The search has retrieved a total of 142 documents having at least one term in common with the query.

Do you want to limit results with any Boolean constraints? (Y/N)
> *N*

How many documents do you want to see?
If you just press RETURN five documents will be displayed
> < *RETURN* >

How much of the data do you want to see:

- Title only type : 1
- Title and Abstract type : 2

> *1*

If at any stage you decide that you don't want to see any more documents just type #.

1/163
Kadri in znanje: Primer splošnoizobraževalnih knjižnic v Ljubljani
* Kolenc, J.: Knjižnica, 32(1988)1/2, str. 23-45

Do you consider this document relevant to your query? (Y/N)
Or if you need to see the abstract as well, type 'A'
> *A*

1/163
Kadri in znanje: Primer splošnoizobraževalnih knjižnic v Ljubljani
* Kolenc, J.: Knjižnica, 32(1988)1/2, str. 23-45

Poleg ustreznih prostorov in sodobne opreme so kakovostni kadri (strokovni delavci) bistveni pogoj za učinkovito izvajanje knjižnično-informacijske dejavnosti. Tudi KIS v SRS in SFRJ namenjena temu vprašanju poseben pomen. Kadrovska funkcija v splošnoizobraževalnih knjižnicah pa ni le potentia temveč tudi agentia (agens) vsakršnega razvoja. Zahtevani prehod v informacijsko družbo postavlja tudi pred nas imperativ: profesionalizacija! Znanje se v knjižničarstvu lahko pretaka le skozi visokoizobražen in kvalificiran kader.Razvoj višjih oblik delitve dela je pogoj za dvig družbenega ugleda (statusa) knjižničarskega poklica.

Do you consider this document relevant to your query? (Y/N)
> *Y*

2/155
Izobraževanje knjižničarskih delavcev v Jugoslaviji
* Berčič, B.: Knjižnica, 31(1987)1, str. 110-129

Do you consider this document relevant to your query? (Y/N)
Or if you need to see the abstract as well, type 'A'
> *Y*

3/185
Začetek visokošolskega študija bibliotekarstva v Sloveniji
* Berčič, B.: Strokovni posvet ZBDS, Bled 1987, 13 str.

Do you consider this document relevant to your query? (Y/N)
Or if you need to see the abstract as well, type 'A'
> *Y*

4/79
Kako je pri nas z razvojem bibliotekarske vede?
* Sepe, M.: Knjižnica, 26(1982)1/2, str. 76-80

Do you consider this document relevant to your query? (Y/N)
Or if you need to see the abstract as well, type 'A'
> *A*

4/79
Kako je pri nas z razvojem bibliotekarske vede?
* Sepe, M.: Knjižnica, 26(1982)1/2, str. 76-80

Razvoj knjižnične stroke naj bi bil preplet treh dejavnikov: Narodne in univerzitetne knjižnice, ki s svojo matično službo kot organizatorjem dejavnosti, z raziskovalnim centrom kot središčem raziskovalnega dela in ob sodelovanju strokovnjakov z drugih specialnih področij bibliotekarske stroke predstavlja nosilca razvoja bibliotekarstva (to NUK-u nalaga tudi Zakon o knjižničarstvu). Skrb za preučevanje in razvoj posameznih strokovnih področij naj še naprej ostane pri Društvu bibliotekarjev. Tretji dejavnik, ki prispeva k razvoju stroke, pa naj bi bile tudi knjižnice same, predvsem s strokovnim izobraževanjem svojih kadrov.

Do you consider this document relevant to your query? (Y/N)
> *Y*

5/204
Trženje v visokošolski knjižnici
* Češnovar, N.: Knjižnica, 33(1989)3/4, str. 123-128

Do you consider this document relevant to your query? (Y/N)
Or if you need to see the abstract as well, type 'A'
> *A*

5/204
Trženje v visokošolski knjižnici
* Češnovar, N.: Knjižnica, 33(1989)3/4, str. 123-128

Marketing oziroma trženje je proces upravljanja, ki ga enačimo s pričakovanji, zahtevami in zadovoljitvijo strank, pri tem pa imamo dobiček. Knjižnica, ki se tržno obnaša, mora: zagotoviti hiter dostop do svojega gradiva; izvajati medknjižnično izposojo, kreirati informacijske zbirke, ki še ne obstajajo; slediti mora modernemu razvoju informacijskih služb. Da lahko sledi tem zahtevam, morajo biti kadri v visokošolskih knjižnicah primerno usposobljeni, da lahko ugotovijo in strokovno presodijo, kaj uporabnik potrebuje. Na tej stopnji pa informacijsko delo že začne preraščati v raziskovalno dejavnost.

Do you consider this document relevant to your query? (Y/N)
> *N*

At this point, if insufficient numbers of useful documents have been obtained, the user can inspect more documents, do a relevance feedback search based upon the relevance judgements given to the system, or carry out a Boolean search.

5.8 Conclusions

In this chapter we have described INSTRUCT, an interactive computer program which demonstrates some of the techniques which have been suggested for implementing online bibliographic retrieval systems. The incorporation of a stemming algorithm into a Slovene version in INSTRUCT represents an important step towards the introduction of non-conventional searching techniques into Slovene information retrieval systems. However, INSTRUCT cannot be introduced into the Slovene information retrieval environment without obtaining extensive evaluation results of its retrieval performance. A description of the experimental test which was used to assess its performance is therefore presented in the next chapter.

Chapter 6

Evaluation of the Stemming Algorithm for Slovene IR – Experimental Environment

6.1 Introduction

Conclusions from previous chapters, mainly Chapter 4 and Chapter 5, can serve as a starting point for designing and conducting experimental tests on the retrieval performance of automatic word conflation in Slovene IR systems. These conclusions have indicated that statistically-based techniques for information retrieval could also be applied to the process of searching documents in Slovene language databases. However, in order to prove this assumption, an experimental test has to be carried out.

Although there are some exceptions—see, for example, the article on using non-conventional retrieval techniques in German language-based IR systems (Fuhr, 1990), the article on the Finnish stemming algorithm (Jäppinen *et al.*, 1985), and the article on IR experiments based on a French set of queries and documents (Chiaramella and Defude, 1987)—experiments on the performance of statistically-based techniques have been so far mainly carried out on English test collections (see, for example, Lennon *et*

al., 1981; Harman, 1987). Therefore, the main problem to be investigated is contained in the following question: *are non-conventional, statistically-based techniques of document retrieval applicable also to the Slovene language*? This question is of particular importance because of a requirement for a *multi-lingual* approach to document retrieval in Slovenia.

In order to test the correlation between non-conventional, statistically-based techniques for document retrieval and the characteristics of the Slovene language, two experiments were carried out. The first experiment, referred to subsequently as *Experiment I*, was concerned with testing the retrieval performance of automatic word conflation in Slovene. Automatic stemming was compared with two other types of text representation, i.e., manual right-hand truncation, carried out by a trained intermediary, and non-stemming. The results obtained from *Experiment I* are described in Chapter 7. The second experiment, referred to subsequently as *Experiment II* and described in Chapter 8, deals with the multi-lingual approach to statistically-based IR methods and techniques. On the basis of performance results from two document collections, the first written in Slovene and the second an English translation of the Slovene texts, an experiment was carried out to test whether statistically-based methods of information retrieval can be successfully applied to two different languages.

In this chapter, the following main components of the test environment for *Experiment I* will be described. First, the question of whether a laboratory or an operational test should be employed will be considered. The following section on test collections will outline their three components: document collection, a set of queries, and relevance assessments. Then, the implementation of three different text representation modules within the INSTRUCT package will be discussed. The final section will describe the methods which were used for the analysis of collected data.

6.2 Laboratory versus operational tests

As pointed out above, the main objective of *Experiment I* was to test whether automatic word conflation can be implemented in Slovene IR systems without any average loss of

retrieval performance, thus allowing users easier access to the systems. This objective relates directly to the most difficult experimental dilemma in IR: what kind of test should be carried out? Should it be a *laboratory* or an *operational* test?

Cleverdon (1966) made this distinction clear in the Cranfield 2 experiments. Operational tests normally involve an evaluation of an existing system; laboratory tests attempt to advance knowledge about individual variables of information retrieval. The idea of conducting experiments under laboratory conditions is to control all variables as far as possible (Robertson, 1981) and, according to Bawden (1990), experiments of this kind are carried out only in order to test some hypothesis which the experiment will prove or disprove.

Although *laboratory-style* evaluations of IR systems have recently been under increasing criticism, particularly from the advocates of *user-oriented* evaluations (UOE)—see, for example, Ellis (1990), Bawden (1990)—the following reasons dictated the application of a laboratory-type evaluation in *Experiment I*:

- the experiment was not concerned with the performance of the complete IR system (i.e., INSTRUCT), but only with one of its sub-systems, i.e., the retrieval performance of automatic word conflation using a single searching strategy;
- the only variable under investigation was the stemming algorithm and its influence on retrieval performance;
- the experiment was based on the simple comparison of automatic word conflation with manual right-hand truncation and non-conflation;
- no user-oriented variables (e.g., human factors) were taken into account as a part of this experiment.

Manual right-hand truncation and automatic stemming have the same purpose, i.e., to improve retrieval performance, in particular its *recall*. The Slovene stemming algorithm can therefore be evaluated by comparing its performance results with the results obtained by the application of manual right-hand truncation. Or, in other words, a per-

formance at least equivalent to the manual right-hand truncation means that stemming can be automated.

Experiment I was accordingly concerned with three different types of text representation (automatic stemming, manual right-hand truncation and non-conflation) and their influence on retrieval performance, expressed in terms of recall and precision. In order to carry out experimental testing, the following procedures were required:

1. design and development of a test collection;
2. implementation of three different modules of query and document processing within INSTRUCT;
3. searching of documents in a database;
4. evaluation of results.

As noted above, this chapter will be concerned mainly with the first two items.

6.3 Test collection

It has been widely accepted in laboratory-type IR experiments that test collections involve three aspects:

- a collection of documents;
- a set of search requests, i.e., queries;
- relevance judgments relating the search request to the document.

The literature on IR experiments reveals that most of the existing test collections were developed for searching documents written in the English language. A comprehensive list of these test collections can be found in Sparck Jones and Bates (1977). Despite many sorts of criticism of these test collections—in particular on the problem of relevance assessments (see, for example, Ellis, 1990; Bawden, 1990)—the existing

test collections have been demonstrated to be very useful tools for various IR experiments. For example, Harman (1987) tested different stemming algorithms using three test collections.

Since *Experiment I* was defined to be a laboratory-type of evaluation, an appropriate Slovene test collection, consisting of Slovene documents, queries and relevance judgments was required. However, the lack of any experimental work in information retrieval and a deficit in evaluating existing IR systems in Slovenia has consequently resulted in the fact that the test collection designed as part of *Experiment I* will be the first such collection in Slovenia. A description of components of this test collection is presented below.

6.3.1 Documents

According to Robertson (1981) a *document set* is, in theory at least, taken as more or less synonymous with text in linguistics, i.e., it describes any piece of linguistic material that can reasonably be considered as a unit (e.g., a scientific paper). For the purpose of evaluating a stemming algorithm it is of particular importance to have a set of documents consisting of terms in natural language, i.e., illustrating the morphology of a particular language. Thus, apart from a title, at least an abstract should form a unit in a document collection.

The design and development of a Slovene document collection was described in a previous chapter. The following is an example of a unit in this document collection.

> Knjižnično-informacijska dejavnost v Sheffieldu, Velika Britanija: vtisi s strokovnega izpopolnjevanja
>
> * Popovič, B.: Knjižnica, 34(1990)1/2, str. 85-100
>
> Prispevek opisuje delovanje in povezovanje knjižnic in informacijskih centrov v Sheffieldu, Velika Britanija. Po opisu nekaterih najzanimivejših knjižnic in informacijskih centrov v Sheffieldu se dotakne predvsem naslednjih tematskih sklopov: obdelave gradiva in organizacije dela v knjižnicah, avtomatizacije, povezanosti knjižnic in informacijskih centrov z družbenim okoljem ter povezovanje knjižnic in informacijskih centrov v regionalnem okviru. Nekaj besed je tudi o reševanju finančne problematike v knjižnicah, kar bo tudi za nas verjetno vedno bolj aktualno.

The size of the test collection in terms of number of documents (i.e., 504 units which represent all available articles in the two journals) is relatively small. However, the nature of the experiment, i.e., a simple comparison of three different types of text representation, and the limited number of variables to be controlled, were sufficient arguments for a decision about the size, form, and the subject coverage of the document collection.

6.3.2 A set of queries

According to Tague (1981), a *query* is the verbalized statement of a user's need and this is often expressed as a short natural language question or statement and may be accompanied by terms chosen manually from an indexing language. So far, great variability has been exhibited by IR experiments in their methods of obtaining queries. The following are some of the issues that must be considered in the process of obtaining test queries:

- The searcher must be correctly aware of the user's requirements, and, therefore, the query must be properly defined and covered (Salton, 1975). Tague (1981) adds that any unclear queries should be rejected;
- There is much variation shown by IR experiments in their method of obtaining requests; one may either solicit the co-operation of the actual users of a system or use queries which are in some sense artificial but under greater control of the investigator.
- Ideally, users should be randomly selected from a pool by the investigator but this is rarely possible as users are normally self-selected because of the degree of cooperation required of them.

These issues about designing a set of queries in IR experiments have served as a starting-point in query development for this experiment.

Query development

In this experiment, eight (8) different persons were used to generate a total of forty-eight (48) different search requests in the librarianship and information science field. This is a reasonable number to allow for at least some assurance that the results obtained are not simple artifacts of an inadequate query set (for more information on sample sizes see Robertson, 1990). A list of all of the queries can be found in Appendix B.

The persons were selected on the basis of research specialization, availability, and willingness to judge the relevance of documents retrieved. Each person was familiar with the library and information science field, either being a librarian himself or a researcher in library science. Each person was asked to produce six requests that were either of his interest or might be asked by library and information science researchers. To aid in the query generation, a detailed and carefully drawn set of instructions was distributed to the group of query authors. The main criteria proposed for the query formulation are similar to those designed by Lesk and Salton (1969) and are presented in Table 6.1.

Table 6.1 shows that each query was expected to represent a real information need, and had to be expressed in grammatically correct and, hopefully, unambiguous Slovene. Positive formulations were required and the queries were to be generated independently from the document collection; in particular, no *source* document was to be used for the formulation of any of the queries. In addition, no limits were defined as to the numbers of words or sentences in each query.

Despite detailed instructions on how each query should be formulated, the nature of a document collection, in particular its small size, required additional interviews with each of the requesters. These interviews were needed because of the following:

- some of the queries covered very specific topics and their processing could result in retrieving a very small number of relevant documents;
- some of the requesters provided similar queries, i.e., covering the same topic;
- while some queries were enhanced with synonyms and abbreviations, other queries

Positive criteria for query formulation	Negative criteria for query formulation
1. Generate queries of real interest to a potential researcher	1. Avoid "exotic" topics and doubtful subject matters
2. Formulate queries in clear, coherent and grammatically correct sentences	2. Do not submit queries corresponding to the contents of a specific document
3. Use positive formulations stating what subject areas are actually wanted	3. Avoid negative formulations, introduced by "except", "not", "other than", etc.
4. Use homogeneous query formulations representing a single topic	
5. Use only common abbreviations	

Table 6.1: Principal criteria for query formulation

were short, simple statements.

Thus, the additional interview helped towards a formulation of the search topics which were clear and meaningful. Since some of the original queries were altered (for example, to cover a broader subject area or to include additional keywords, phrases or common abbreviations) the existing set of 48 queries can therefore be defined as a combination of real and structured queries. Table 6.2 shows the main characteristics of this set of queries before and after stopwording has been carried out.

Before deletion of stop-words, the largest query consisted of 19 terms and the smallest query of only 2 terms. The median number of terms per query was 8. Since queries were formulated mainly as natural language sentences it is interesting to see their quantitative characteristics after the removal of stop-words. The total number of words was reduced from 370 to 293, and the median number of terms per query was 6.

In order to carry out a comparative evaluation of automatic stemming and manual

Quantitative characteristics	Before stopwording	After stopwording
Number of queries	48	48
Total number of terms in a set of queries	370	293
Average number (median) of terms per query	8	6
Maximum number of terms in a query	19	15
Minimum number of terms in a query	2	2

Table 6.2: The main characteristics of the set of queries before and after deletion of stop-words

right-hand truncation, all queries from the test set were also processed by a *trained intermediary*. The every-day job responsibility of this person is to search information from various English and Slovene databases. Since all Slovene document databases can be described as descriptor-based collections, the processing of natural language queries in Slovene was a new and demanding linguistic experience for the intermediary. Different tools (linguistic handbooks, dictionaries) were therefore used as a help during manual processing of queries.

Using knowledge about Slovene linguistics, the professional intermediary thus truncated all terms (apart from stop-words) manually on the right with the objective of achieving good retrieval results. For example, if the query consisted of the following statement:

> Karkoli o Narodni in univerzitetni knjižnici (NUK) v Ljubljani
> (Anything about the National and University Library in Ljubljana)

the professional intermediary first excluded stop-words (i.e., KARKOLI, O, IN, V) and then formulated the query, using right-hand truncation. The result of his manual involvement was a list of truncated terms, as shown in the example below:

> Narodn? univerz? knjižnic? NUK? Ljubljan?

A list of all queries manually reformulated by the professional intermediary can be found in Appendix C.

6.3.3 Relevance assessments

Most commonly, documents output by the system in IR experiments are individually assessed for relevance to the user's need. The word *relevance*—as pointed out by Robertson (1981)—has been used in many different ways, but broadly corresponds to the question: how well does the document match the user's need?

There is a wealth of literature relating both to relevance in general and to obtaining relevance assessments when setting up a test collection in particular. First of all, there is no doubt that relevance is a subjective notion, i.e., different users may differ about the relevance or non-relevance of particular documents to a given question. However, van Rijsbergen (1979) suggests that the difference is not large enough to invalidate experiments which have been made with test collections for which requests with corresponding relevance assessments are available.

One major study has investigated the factors that influence relevance decisions (Cuadra and Katter, 1967). This shows that relevance judgments are influenced by many factors (e.g., the skills and attitudes of the judges, the documents used). Thus, relevance is an imprecise concept, always dependent on the precise situation of the user (Bawden, 1990).

However, since the performance measures in IR experiments rely on being able to distinguish useful and required items from those which are not so useful, some type of relevance assessment is required. Lesk and Salton (1969) have found that inconsistency of relevance assessments may have no effect on certain aspects of system evaluation. In fact, Robertson (1981) suggests that assessment of relevance allows a *harder* form of analysis than any other assessments in this category.

Having in mind the vague definition of relevance and its subjective notion there are many practical problems arising in obtaining relevance judgments. Tague (1981) points out that actually getting assessments of document relevance is an even greater problem

than getting queries. Robertson (1981) listed the following aspects of the process of obtaining relevance assessments:

1. Who is to make the assessments? Robertson (1981) states that where the request is stimulated by a genuine information need, the requester should decide on relevance. This may cause problems since the user may not be prepared to assess as many documents as desired by the experimenter. Many experiments rely on third parties for assessments though this is regarded with distrust.

2. How much of the document should the relevance judge see before making a judgment? Ideally, the entire text of the document should be produced but usually titles and abstracts are used. Titles alone are very poor indicators.

3. Which documents should be judged? Ideally, the whole document collection should be assessed but as this would usually be impossible (because of the size of the document collection under consideration), the set to be judged will often consist of the pooled output of various searches.

4. What instructions should be given to the judges, and in what form should the assessments be obtained? It is important that all individuals making relevance assessments receive the same instructions. It has frequently been pointed out that relevance embodies two distinct notions, i.e.:

   (a) is the document about the subject of the query? (i.e., the *aboutness* criteria);

   (b) will the document be useful to the user? (i.e., the *usefulness* criteria).

   Users should, therefore, be clear whether they are assessing *subject relevance* or *pertinence*. In the former usage a relevant document is simply one which deals (to a greater or lesser extent) with the same subject matter as that of the query, whereas for a document to be pertinent it has to contain information which is new and useful to the originator of the query in the subject area of the query.

   In addition, more than two categories (i.e., relevant/non-relevant) of relevance assessments are often provided to the users, although there are not any experimental

performance measures that consider anything other than relevant/non-relevant. These additional degrees can be used only in the qualitative type of evaluation. At the analysis stage, they are conflated into just two categories. However having more than two degrees of relevance makes it easier for a judge to give a relevance value to documents in a collection.

The practical problems as outlined above were considered in detail when requesters were asked to judge documents for relevance as part of *Experiment I*. The relevance assessments were carried out in the following manner:

1. Relevance assessments of a set of retrieved documents were made by the requester, i.e., user; this means that no third party (for example, a team of experts in librarianship) was involved in this part of the experiment.

2. Requesters judged documents for their relevance on the basis of information obtained from the title and abstract of each retrieved document.

3. The set of documents to be judged for relevance consisted of the pooled output of three different types of search (i.e., automatic stemming, right-hand truncation, non-conflation), using the ranked-output cutoff procedure. The cutoff point was 10, i.e., only the first ten retrieved documents from each type of search were considered further.

4. All relevance judges were given the same instructions about carrying out relevance assessments. In the experiment reported here relevance was used with the meaning of *aboutness*, and not *usefulness*. A document was therefore judged to be relevant only

   > "... if it is directly stated in the abstract as printed, or can be directly deduced from the printed abstract, that the document contains information on the topic asked for in the query" (Lesk and Salton, 1969).

   Consequently, some of the documents were therefore still judged to be relevant, although:

   (a) the requester has already read these documents,

(b) the requester does not usually read documents written by a particular author;

(c) the document is obsolete, etc.

5. Since this experiment was interested mainly in quantitative evaluation, the requesters were instructed to supply only binary (yes or no) relevance assessments.

To summarize, the experiment reported here was using results from previous projects concerned with obtaining relevance assessments as a part of a test collection. The noticeable deviation from the "ideal" path can be found in two questions, i.e., which documents should be judged, and how many degrees of relevance assessments should be obtained from the users? A decision to use only a retrieved pool of documents, and to apply only yes/no criteria was based on the nature of this testing. The aim of the experiment was not to evaluate the whole system, but to find basic performance differences between three types of text representation. Since the project did not contain many variables to be controlled, the simplicity of its design was of crucial importance.

Having obtained relevance assessments on the retrieved pool of documents, it became possible to measure the retrieval performance of the three different types of search. However, in order to carry out such an evaluation, three different modules of query and document processing had to be implemented within an information retrieval system. Since INSTRUCT was selected as a test bed for this project, the incorporation of these modules in INSTRUCT is described in the next section.

6.4 Text representation modules in INSTRUCT

In order to carry out the test, the following three different modules were developed within the INSTRUCT package:

- automatic stemming,
- manual right-hand truncation,
- non-conflation.

The main characteristics of each module and its implementation within the INSTRUCT package are described in the section below.

6.4.1 Automatic stemming

After stop-words are removed from both a query and documents in a database, suffix removal is carried out, using the stemming algorithm as described in Chapter 4. Then, a best-match search is carried out, achieving the ranking of documents. The best-match searching algorithm and weighting techniques implemented in INSTRUCT were described in detail in Chapters 1 and 5.

Suppose we want to search for documents containing a word ARHIVI. The inverted file of INSTRUCT consists of data as illustrated in Table 6.3.

Dictionary file	Postings file
2 ARAB	154,166
2 ARGUM	242,361
3 ARHITEK	128,381,390
19 ARHIV	17, 24, 30, 66, 77,116,219, 220,225,234,237,249,296, 306,315,324,331,358,389
1 ARIS	166
1 ARISTOK	387
1 ARISTOT	416
1 ARMAR	124
1 ARTIKUL	433
1 ARTOT	162

Table 6.3: An excerpt from the inverted file of INSTRUCT (stemmed version)

It can be seen from Table 6.3 that the word ARHIVI must be truncated to ARHIV*, in order to retrieve its morphological variants. The actual searching then results in a list of 19 document identifiers. Or, in other words, using the main part of the term weighting formula, i.e.,

$$log\left(\frac{N}{n_i}\right) + log\ (9.0)$$

where:
N – number of documents in a collection,
n_i – number of documents indexed by i^{th} term,
9.0 – a value of the constant C, as defined by Croft and Harper (1979) and also explained in Chapter 1,

then the elements of the array denoting the similarity between the query and each of the documents are incremented by

$$log\left(\frac{504}{19}\right) + log\ (9.0)$$

6.4.2 Non-conflation

This module is in fact responsible only for the removal of stop-words from both documents and queries. After stop-words are deleted, the remaining words remain intact. It means that the part of the inverted file of INSTRUCT concerned with the word ARHIVI is as shown in Table 6.4.

The first conclusion which can be drawn from Table 6.4 is that the dictionary file of the unstemmed version of INSTRUCT consists of a larger number of terms than the dictionary file in the stemmed version. It can be seen above that there are in fact 15 variants of the stem ARHIV*.

Since the non-conflated representation of text means that words from queries and documents remain unchanged, searching for documents containing the term ARHIVI results in only 2 document identifiers. The elements of the array within term weighting module are therefore incremented by

$$log\left(\frac{504}{2}\right) + log\ (9.0)$$

6.4.3 Manual right-hand truncation

The module performing manual right-hand truncation within INSTRUCT exhibits the following features:

- the same inverted file is used as for the unstemmed version of INSTRUCT (see Table 6.4);

Dictionary file		Postings file
1	ARHITEKTURE	390
4	ARHIV	116,296,358,389
1	ARHIVA	315
2	ARHIVI	30,225
1	ARHIVIRANJU	324
1	ARHIVISTI	225
2	ARHIVISTIKA	234,249
1	ARHIVISTIKE	331
2	ARHIVISTIKO	77,331
1	ARHIVOM	237
6	ARHIVOV	219,220,225,296,306,331
1	ARHIVSKE	331
2	ARHIVSKEGA	225,331
5	ARHIVSKIH	17, 24, 66,225,331
1	ARHIVSKIMI	296
1	ARHIVU	358
1	ARIS	166
1	ARISTOKRACIJE	387
1	ARISTOTEL	416
1	ARMARIJEV	124

Table 6.4: An excerpt from the inverted file of INSTRUCT (unstemmed version)

- words from the query are manually processed by a trained intermediary in two stages:
  - removal of stop-words from the query,
  - suffix removal from remaining words.

Thus, if a trained intermediary truncates a searching term ARHIVI to ARHIV*, a dictionary file is searched to find various lists identified by a string ARHIV. Table 6.4 indicates that there are 15 such lists. These lists are then OR-ed together, and x (where x = number of distinct identifiers) is counted. Finally, the elements of the array within the term weighting module are increment by using the following formula,

$$log\left(\frac{N}{x}\right) + log\ (9.0)$$

or, in this experiment, where $x = 19$,

$$log\left(\frac{504}{19}\right) + log\ (9.0)$$

Thus, in this particular case, both stemming and manual right-hand truncation would have precisely the same effect.

To summarize, there are three main differences between the three types of text representation in INSTRUCT: the size of the dictionary file; the role of an intermediary; and the number of records retrieved. The first difference relates to the size of the dictionary file, as illustrated in Table 6.5, which demonstrates that the employment of the Slovene stemming algorithm, eliminating morphological variants of terms, results in a large compression of the dictionary file. The percentage of this compression (65.6%) is in accordance with the complex Slovene morphology and is even higher than that obtained in tests on a Slovene text corpus, as described in Chapter 4. In other words, the notion about the effect of the size of the database on the level of compression has again been confirmed. While the application of the stemming algorithm on the text corpus consisting of 2,616 distinct word types resulted in a 54.7% level of compression, its employment on the larger dictionary file (i.e., 8,602 distinct word types) produced a 65.6% level of compression.

Text representation	Number of terms in a dictionary file
Automatic stemming	2,957
Truncation	8,602
Non-conflation	8,602

Table 6.5: Frequency characteristics of dictionary files in the three different versions of INSTRUCT

The next difference can be found within the query input module. While stemmed and unstemmed versions of INSTRUCT allow natural language input, manual right-

hand truncation requires the active role of a trained intermediary who has to conflate words at the right point before inputting a query.

However, the largest difference between these three modules of text representation relates to the processing of a query within best-match searching. While automatic stemming and manual right-hand truncation allow retrieval of morphologically related terms, the non-conflated module can retrieve only those documents which exactly match words from the query. For example, searching for documents containing a word ARHIVI will result in a non-stemmed version of INSTRUCT in a list of only two documents; both other modules (i.e., automatic conflation and manual right-hand truncation) are able to retrieve a much larger number of relevant documents.

There is no doubt that the differences outlined above between the three modules of text representation will have a large effect on the retrieval performance in this experiment.

In order to obtain comparable performance results from the three different types of text representation within INSTRUCT, a single search strategy, i.e., *best-match searching*, has been used in this experiment. Best-match or nearest neighbour searching is described in detail in Chapter 1. In addition, a description of its implementation within INSTRUCT can be found in Chapter 5 (see also Hendry *et al.*, 1986a,b).

6.5 Methods for the analysis of data

Most information retrieval tests are ultimately concerned with the effectiveness of each system. In essence, the question of measuring effectiveness is simple: we want to decide *how well* the information system is operating, compared with some theoretical maximum (Bawden, 1990). In an operational situation, this means how well it meets the real needs of its users. In an experimental setting, some more or less artificial measures must be adopted as a surrogate for user satisfaction.

In a laboratory setting the measures of performance used in the majority of retrieval tests are the well-known factors of *recall* and *precision*. These two measures of retrieval

effectiveness have been at the forefront of retrieval evaluation since Cranfield; they are defined using the following formulae:

$$RECALL = \frac{number\ of\ relevant\ documents\ retrieved}{total\ number\ relevant\ documents\ in\ collection}$$

$$PRECISION = \frac{number\ of\ relevant\ documents\ retrieved}{total\ number\ documents\ retrieved}$$

Recall is therefore a measure of how well the system performs at yielding up all the relevant items within it, and precision measures how well the system provides only the relevant items. One of the main achievements of the Cranfield tests was the experimental demonstration that recall and precision are inversely related (Bawden, 1990).

These performance measures, of course, depend on the relevance assessments of documents. We have already discussed the impreciseness of the concept of relevance. This concept also assumes that all of the relevant items are equally useful. However, though their simplicity may be criticized, recall and precision are undoubtedly significant for practical evaluation. As Cleverdon (1966) states: "The unarguable fact however is that they are fundamental requirements of the users, and it is quite unrealistic to try to measure how effectively a system or subsystem is operating without bringing in recall and precision".

Since this experiment was concerned with a simple comparison of three different types of text representation in queries and documents, recall and precision were found to be appropriate measures of retrieval performance. Therefore, to answer the question of which type of text representation performs the best, the results of searching were analyzed in the following stages (the term "system" here corresponds to a particular type of text representation within INSTRUCT):

1. for each query and system, recall and precision were used as measures of the system's response to the request;

2. for each system, these measures were averaged over the query set;

3. the averages for different systems were compared;

4. the statistical significance of the differences between the systems were tested using the sign test and the Kendall coefficient of concordance, W.

It has to be noted at this point that *relative* recall was calculated, based on the substitute list of total relevant items in the collection, i.e., a pool of retrieved documents. The true value of recall can only be calculated, if someone examines each and every item in the collection, to see if it is relevant. This was not the case in *Experiment I* in which a pool of retrieved documents was used as a substitute for the complete document collection.

Perhaps it should also be pointed out that the precision values were rather superfluous since, with a ranked-output cutoff procedure, all the available information is contained in the recall figure – at a particular cutoff point if the recall figure for one system is better than that for another then the precision figure is similarly so.

Although *Experiment I* was concerned mainly with a quantitative analysis of data, in particular using a measure of recall and precision, some other trends in IR experiments were also taken into account. It is interesting to note that recent literature (see, for example, Bawden, 1990) emphasizes the development of a small set of queries which can then be used for a detailed qualitative analysis. An example of such an approach is a method known as *failure analysis*, which dates back to Lancaster's early MEDLARS experiments (Lancaster, 1969) and afterwards employed in many information retrieval tests (see, for example, McCain *et al.*, 1987; Harman, 1991, etc.). Its main purpose is to find out *why* things work as they do, and how matters may be improved. Sparck Jones (1981) notes that failure analysis "...is not part of an experiment proper, but makes a very important contribution to the broader study of retrieval system behaviour." The main question this type of analysis tries to answer is *why* certain relevant documents were not retrieved, or *why* certain non-relevant documents were retrieved. On this basis, recommendations for real improvements in systems can be made.

Because of this trend it was decided that *Experiment I* should also be concerned—although on a very small scale—with the question of why processing of some queries

resulted in better performance achieved by the manual right-hand truncation than automatic word conflation and *vice versa*.

6.6 Conclusions

In *Experiment I*, characterized as a laboratory-type test, the following items were employed:

- a test collection;
- three different types of text representation (automatic stemming, manual right-hand truncation, non-conflation);
- a best-match searching strategy implemented in the INSTRUCT retrieval package.

It has to be noted that this test collection represents the first such collection to be set up in a Slovene IR environment. The implementation of these items can serve as a starting point for a decision on how to collect data and how to present the results of *Experiment I*. These issues are detailed in the next chapter.

Chapter 7

Evaluation of the Stemming Algorithm for Slovene IR – Analysis of Results

7.1 Introduction

The main objective of *Experiment I* was to test the following two hypotheses in the context of a best-match environment:

- *HYPOTHESIS 1*: There is a significant difference in retrieval effectiveness between queries which have been subjected to automatic word conflation and those which have not been stemmed.
- *HYPOTHESIS 2*: There is no significant difference in retrieval effectiveness between queries which have been subjected to automatic word conflation and those which have been submitted to manual right-hand truncation, carried out by a trained intermediary.

So far, there has been no published evidence about similar experiments in Slovene IR systems. It was also hoped that results of *Experiment I* would provide a framework to

test a multi-lingual approach to IR, using statistically-based techniques.

This chapter will first describe how data was collected. On this basis, the analysis of results will be presented.

7.2 Collection of data

To obtain the data required for the evaluation, the following procedures were employed:

- searching across a document collection, using three different types of query processing;
- retrieving a pool of document records that were then judged relevant or non-relevant by requesters;
- analyzing the obtained data, using basic measures of retrieval performance (i.e., recall and precision).

These procedures are described below.

7.2.1 Searching

On the basis of two sets of queries (i.e., a set of queries written in natural language, and a set of queries processed by a trained intermediary) the following three different types of searching were carried out for each query:

- best-match searching using automatic word conflation;
- best-match searching using manual right-hand truncation;
- best-match searching using unstemmed words.

The execution of these search strategies yielded three sets of retrieved documents for each query. Since the test collection consisted of 48 different queries, a total of 144 listings of retrieval output were produced.

7.2.2 A pool of retrieved documents

Since a ranked output is provided by best-match searching, it is important for evaluation purposes to establish rank cutoff points. Having in mind the size of the document collection and the set of queries, a cutoff was defined at position 10. This means that, after INSTRUCT ranks documents in order of decreasing similarity with the query, only the top 10 documents are examined further.

Using this ranked-output cutoff procedure, a pool of retrieved documents was developed. Or, more precisely, after each of the 48 queries was processed in three different ways, the first 10 documents of each search formed the pooled output. Thus, each query had an associated pool of retrieved documents.

The list of pooled documents for each query was returned to the requester for relevance assessments. Relevant documents were then compared with the top 10 documents from each type of search. This comparison was the starting-point for measuring the performance of the three different types of search.

7.3 Analysis of results

7.3.1 Recall and precision as measures of retrieval effectiveness

On the basis of a comparison of the relevant items from the retrieved pool with the top 10 documents in the three different lists, the absolute figures were obtained for each query, and then recall values calculated (see Table 7.1). The precision values, as a percentage, can be extracted from absolute numbers if multiplied by 10. All these values were then also averaged over the request set, for each type of search separately, in order to obtain the mean number of relevant documents, i.e., the aggregate precision and the aggregate recall.

Table 7.1 shows that automatic stemming and manual right-hand truncation employed in best-match searching achieved almost the same results. While the automatic stemming resulted in a total of 302 relevant documents over a set of queries (or, in

Query	Stemmed		Truncation		Unstemmed		Pooled ret/rel	Total distinct retrieved
	rel	%	rel	%	rel	%	no. docs	no. docs
1	6	100	5	83	4	67	6	17
2	9	90	9	90	4	40	10	17
3	9	100	7	78	5	56	9	18
4	7	100	7	100	4	57	7	15
5	8	89	6	67	6	67	9	17
6	8	89	8	89	2	22	9	20
7	10	77	10	77	6	46	13	17
8	3	100	3	100	1	33	3	17
9	7	58	10	83	3	25	12	19
10	7	78	6	67	2	22	9	18
11	9	64	9	64	8	57	14	18
12	10	67	10	67	7	47	15	18
13	8	89	8	89	8	89	9	15
14	9	82	9	82	7	64	11	15
15	9	82	9	82	9	82	11	12
16	5	83	5	83	3	50	6	17
17	7	100	6	86	6	86	7	16
18	7	100	6	86	6	86	7	14
19	6	43	6	43	8	57	14	20
20	10	50	9	45	10	50	20	21
21	10	83	10	83	3	25	12	19
22	6	86	6	86	3	43	7	17
23	6	86	6	86	4	57	7	16
24	3	75	4	100	1	25	4	21
25	6	50	7	58	7	58	12	19
26	7	87	7	87	6	75	8	15
27	8	80	8	80	8	80	10	13
28	7	54	8	62	8	62	13	18
29	5	62	5	62	4	50	8	17
30	1	100	1	100	0	0	1	16
31	5	62	5	62	4	50	8	19
32	3	60	3	60	2	40	5	17
33	5	100	5	100	3	60	5	11
34	4	67	4	67	3	50	6	18
35	10	83	8	67	5	42	12	19
36	9	100	9	100	5	65	9	15
37	3	75	2	50	1	25	4	22
38	4	67	3	50	4	67	6	20
39	7	78	8	89	3	34	9	19
40	5	83	4	67	4	67	6	18
41	3	37	3	37	5	62	8	20
42	2	100	2	100	0	0	2	17
43	6	100	4	67	2	34	6	21
44	5	62	6	75	5	62	8	17
45	6	100	6	100	2	34	6	16
46	2	50	3	75	3	75	4	20
47	5	71	5	71	3	43	7	18
48	5	71	7	100	3	43	7	21
Total	302	75	297	74	210	52	401	840

Table 7.1: Number of relevant retrieved documents and aggregate recall for 48 queries, using three different types of search (cutoff 10)

other words, 6.3 relevant documents per query), the manual right-hand truncation performed just slightly worse, in that it retrieved in total 297 relevant documents (or, in other words, 6.2 documents per query). On the other hand, non-conflation of terms in queries and documents resulted in a very poor performance. A set of only 210 relevant documents was retrieved by this type of search, or, in other words, only 4.4 documents per query.

The above data can also be interpreted in terms of retrieval effectiveness, using the measure of *recall*. For each query and for all queries in aggregate, recall percentage was calculated as:

$$\frac{\textit{retrieved and relevant in database}}{\textit{pooled retrieved and relevant}} \times 100$$

Table 7.1 reveals that there is a very little performance difference between automatic stemming and manual right-hand truncation. While the former achieved an average recall of 75%, the latter was slightly behind with an average recall value of 74%. The unstemmed processing of queries and documents performed the worst, achieving an average recall of only 52%.

The *precision* figures can also be extracted from Table 7.1, i.e., using the mean number of relevant documents. Thus, precision for automatic stemming was 63%, for the manual right-hand truncation 62%, and for non-conflation 44%. The average recall and precision values for the three different types of text processing are summarized in Table 7.2.

The results in the tables above indicate the following:

- both types of word conflation (i.e., automatic stemming and right-hand truncation) perform much better than non-conflation; i.e., they are able to retrieve a higher percentage of relevant documents from a database;
- there is a very little difference between those two types of text representation; both the automatic stemming and manual right-hand truncation are able to retrieve a similar percentage of relevant documents from a database.

Retrieval effectiveness	Stemmed	Truncation	Unstemmed
RECALL	75	74	52
PRECISION	63	62	44

Table 7.2: The average recall and precision values of three different types of search (*cutoff 10*)

In order to investigate whether the cutoff factor influenced the performance results, the additional ranked-cutoff procedure at the position 5 (i.e., the top 5 documents) was employed. Performance differences between three different types of search at cutoff 5 are summarized in Table 7.3.

Results	Stemmed	Truncation	Unstemmed
No. of rel.docs.	182	183	138
Mean of relevants	3.8	3.8	2.9
Recall	45	45	34
Precision	76	76	58

Table 7.3: Results from three different types of search at cutoff 5

Using the lower cutoff position (i.e., the top 5 documents)—with the same pool of retrieved documents—means that the average recall numbers in Table 7.3 are lower than in Table 7.2. However, the most important fact is that the performance differences between three types of text representation remain more or less unchanged. This means that automatic stemming and right-hand truncation exhibit a superior performance over unstemmed processing of words, and that the change in cutoff does not affect the relative performance of the three methods.

In order to prove, firstly, that both automatic stemming and manual right-hand truncation perform equally well, and, secondly, that they are superior to unstemmed word processing, a statistical *significance test* on the differences has to be carried out. In this experiment, the following two tests were employed:

- the sign test;
- the Kendall coefficient of concordance, W.

Since no difference was found between results obtained at cutoff 10 and cutoff 5, data from the former cutoff point was used for both significance tests.

7.3.2 Significance tests

The sign test

The *sign test* gets its name from the fact that it is based upon the direction of differences between two measures. The sign test is applicable to the comparison of two related samples. It is particularly useful for research in which it is possible to determine, for each pair of observations, which is the "greater" (Siegel and Castellan, 1988).

In applying the sign test, the focus is on the direction of the difference between a pair, noting whether the sign of the difference is positive (+) or negative (−). In addition, a "tie" (0) occurs when two values are equal. In this experiment, three pairs of observations were investigated for their significance difference:

- the performance of automatic stemming vs. the manual right-hand truncation;
- the performance of automatic stemming vs. non-conflation;
- the performance of the manual right-hand truncation vs. non-conflation.

For each pair, the sign test required the introduction of the null hypothesis (H_0) and alternative hypothesis (H_1). They can be generalized as follows:

- H_0 : a difference between two types of text representation is *zero* (e.g., the application of automatic word conflation and manual right-hand truncation results in a retrieval of a similar percentage of relevant documents);

- H_1 : a difference between two types of text representation is *positive* (e.g., the employment of the automatic stemming results in a larger number of relevant documents than manual right-hand truncation).

Two additional details are required for the sign test, i.e., the significance level (α) and the number of couples (N) under observation. While the significance level (α) was defined at 0.05, the number of couples (N) was equal to the number of queries, i.e., N = 48. If a matched pair showed no difference (i.e., the difference was zero and had no sign), it was dropped from the analysis and N was reduced accordingly.

Since H_1 predicts the direction of the differences, the rejection region was defined to be one-tailed. It consists of all values of x (where x is the number of +s since the prediction for H_1 is that positive differences will predominate) for which the one-tailed probability (p) of occurrence when H_0 is true is equal to or less than $\alpha = 0.05$. If N is still larger than 35, a method for large samples has to be applied (see Siegel and Castellan, 1988), using the following formula:

$$z = \frac{2x \pm 1 - N}{\sqrt{N}}$$

Results of the significance test on these three pairs of observations, using the sign test, are presented in Table 7.4 and discussed below.

Automatic stemming *vs.* manual right-hand truncation. Table 7.4 shows that automatic stemming was more successful in 12 observations and less successful in 8 observations than manual right-hand truncation. The number of tied cases was 28; consequently N was reduced to 20. Appendix Table D in Siegel and Castellan (1988) shows that for $N = 20$ the probability of observing $x >= 12$ has a one-tailed probability when H_0 is true of 0.868. Since this value is not in the region of rejection for $\alpha = 0.05$, the decision is to reject H_1 in favour of H_0. Thus, this test has shown that there is

Comparison	+	−	0	z	p
Stemming *vs.* truncation	12	8	28	−	0.86800
Stemming *vs.* non-conflation	37	5	6	4.78	0.00003
Truncation *vs.* non-conflation	34	4	10	4.70	0.00003

Table 7.4: Frequency distribution of the direction of differences between three pairs of text representation (with z and p values)

no significance difference between automatic word conflation and manual right-hand truncation.

Automatic stemming *vs.* **non-conflation.** It can be seen from Table 7.4 that automatic stemming performs better than non-conflation in 37 cases. Its performance was less successful in only 5 cases, and equal values were obtained in 6 cases. It follows that N has to be reduced to 42. Since N is still larger than 35, a method for large samples was applied resulting in a value of $z = 4.78$. Reference to Table A in Siegel and Castellan (1988) reveals that the probability $z >= 4.78$ when H_0 is true is 0.00003.

Since 0.00003 is smaller than $\alpha = 0.05$, the decision was to reject the null hypothesis in favour of the alternative hypothesis. Thus, the sign test proved that there is a significant performance difference between automatic stemming and non-conflation. Or, in other words, automatic stemming is able to produce significantly better results than non-conflated processing of words.

Manual right-hand truncation *vs.* **non-conflation.** Table 7.4 shows that manual right-hand truncation performs better than non-conflation in 34 cases. Its performance was less effective in only 4 cases, and "ties" were obtained in 10 cases. Consequently,

N was reduced to 38 and the method for large samples used again. Applying the above formula, a value of $z = 4.70$ was produced. Reference to Table A in Siegel and Castellan (1988) revealed that the probability $z >= 4.70$ when H_0 is true is 0.00003.

Thus, the decision was again to reject the null hypothesis in favour of alternative hypothesis. Or, in other words, the manual removal of endings from words in queries contributes significantly to better performance results than non-conflation.

Conclusions. To summarize the results of the above three tests:

- there is no significant difference in retrieval performance between automatic stemming and manual right-hand truncation; both types of text representation are equally successful in retrieving relevant documents in the Slovene document collection;
- there is a significant difference in retrieval performance between word conflation and non-conflation; automatic word conflation and manual right-hand truncation are much more successful in retrieving relevant documents from the Slovene document collection than non-conflation.

However, since only pairs of variables were compared in the sign test, there was a need to introduce another significance test which is able to differentiate among three or more variables. This test is known as the Kendall coefficient of concordance, W.

The Kendall Coefficient of Concordance, W

The use of this test requires k sets of rankings of N objects or individuals. On this basis, the association among them can be determined, using the *Kendall coefficient of concordance, W*. W expresses the degree of association among k such variables, that is, the association between the k sets of rankings. Or, in other words, W measures the extent to which k rankings of the same set of N objects are in agreement with each other (Siegel and Castellan, 1988). The test, therefore, consists of the following stages:

1. to find the value of W, i.e., to determine its significance (when W is significant, then a high agreement exists);

2. on the basis of a high agreement, W can be interpreted (i.e., a decision about the "best estimate" can be made).

Thus, in this experiment, the first step was to determine agreement among users (i.e., 48 queries) on the association among three different types of search. If that agreement exists then the best type of search will be defined. Using a set of 48 queries, the following hypotheses were introduced:

- H_0 : There is no agreement among the users on the association among three different types of search.

- H_1 : There is an agreement on the association among three different types of search.

The significance level (α) was again defined at 0.05 (i.e., $\alpha = 0.05$). To compute W the data is first arranged into a $k \times N$ table with each row representing the ranks assigned by a particular judge to the N objects. These ranks (i.e., a search retrieving the largest number of relevant documents has a rank 1, etc.) are assigned to all 48 queries. On this basis, the sum of ranks, R_i, and the average rank, $\overline{R}_i$, are defined in each column, as shown in Table 7.5.

Type of search	Sum of ranks	Average rank
Stemmed	79	1.6
Truncation	79	1.6
Unstemmed	128	2.7

Table 7.5: The sum of ranks and the average rank for each type of search

To obtain the mean value of R_i, the sum of ranks (286) is divided by N (3); the mean value is equal to 95.3. Each of the R_i may then be expressed as a deviation from the mean value. The larger these deviations, the greater the degree of association among the k sets of ranks. Finally, s, the sum of squares of these deviations is found ($s = 1,600$). Knowing these values, the value of W was calculated, using the following formula (see Siegel and Castellan, 1988):

$$W = \frac{s}{\frac{1}{12}k^2(N^3 - N)}$$

The value obtained for W was 0.347. This value for W was then compared with the critical values to determine whether there is a statistically significant degree of agreement between the k rankings.

To test the significance of the value of W ($W = 0.347$), the probability associated with this value was determined. This was done by first calculating the *Chi Square* value from,

$$\chi^2 = k(N - 1)W$$

Thus, the calculated χ^2 value is 33.3. Referring to the table of critical values of *Chi Square* (Siegel and Castellan, 1988), we find that the probability associated with χ^2 is less than 0.001 ($p < 0.001$).

Since $p < 0.05$ the H_0 can be rejected in favour of H_1. Or, in other words, the value of W shows that there is a high degree of agreement among users on the association among the different types of search. This agreement is much higher than it would be by chance.

It follows that users were applying essentially the same standard in ranking the N objects (i.e., different types of search) under study. To find out which type of search performed the best, the sum or average of ranks can be used. The "best estimate" is associated, in a certain sense, with a least-squares estimate. In this experiment, the best retrieval performance can, therefore, be assigned to automatic stemming and manual right-hand truncation, for in each of their cases the sum of ranks are equal $R_1 = R_2 = 79$ (or, the average sum = 1.6), or, in other words, the lowest value

observed.

To summarize, the significance test, using the Kendall coefficient value of concordance, W, has again demonstrated the following:

- both types of word conflation (i.e., automatic stemming and manual right-hand truncation) perform significantly better in a Slovene IR environment than non-conflation;
- there is no significant difference between automatic stemming and manual right-hand truncation; both types of text representation are able to retrieve a similar percentage of relevant documents from the Slovene document collection.

In this context, it is interesting to emphasize the quite huge performance difference between unstemmed and stemmed (automatic conflation/manual right-hand truncation) processing of words. This difference is much larger than the performance difference obtained from experiments carried out on English document collections (e.g., Lennon *et al.*, 1981; Harman, 1991). For example, the results of experimental tests by Harman (1991) revealed no substantial difference between full word retrieval and retrieval using suffixing. Although individual queries were affected by stemming, the number of queries with improved performance tended to equal the number with poorer performance, thereby resulting in little overall change for the entire test collection.

In order to prove that these performance differences between Slovene and English IR systems reflect the richer Slovene morphology, a similar test to that carried out by Harman (1991) was repeated in the context of *Experiment I*. Two types of best-match search, i.e., a retrieval based on suffixing, and a full word retrieval, were employed on the English database. This database was developed as part of a test collection for *Experiment II*, and is described in detail in Chapter 8. It is important to note that this collection consisted of an English translation of the Slovene text, and therefore contained identical documents and queries as used in *Experiment I*. The employment of unstemmed and stemmed searching on the English database (referred to as ENGL) produced the following results as presented in Table 7.6; these results are compared

with results of the similar search on the Slovene database (referred to as SLOV).

Type of search	SLOV	ENGL
Stemmed	302	248
Unstemmed	210	234

Table 7.6: Number of relevant documents retrieved by stemmed and unstemmed searches on the English and Slovene databases

The results in Table 7.6 clearly show that the employment of the English stemming algorithm produces only a slight improvement over non-conflation (248 relevant documents *vs.* 234 relevant documents). Moreover, the application of the sign test reveals no significant performance difference between stemmed and unstemmed text representation in an English information retrieval system, as evident in Table 7.7.

Comparison	+	−	0	p
Stemming *vs.* non-conflation	17	13	18	0.81900

Table 7.7: Comparison between automatic stemming and non-conflation in the English database (results of the sign test)

Table 7.7 shows that automatic stemming performs better than non-conflation in 17 cases. Its performance was less effective in 13 cases; the number of tied cases was 18. All tied cases were dropped from further analysis; consequently N was reduced to 30. Appendix Table D in Siegel and Castellan (1988) shows that for $N = 30$ the probability of observing $x >= 17$ has a one-tailed probability when H_0 is true of 0.819.

Since this value is not in the region of rejection for $\alpha = 0.05$, the decision was to reject H_1 in favour of H_0. Thus, this test has shown that there is no significant difference between these two types of text representation.

These results are entirely analogous to those presented by Harman (1991). There is no doubt that a language with a rich morphology (e.g., Slovene) correlates with a larger distinction between retrieval performance based on suffixing, and retrieval performance based on using full words. This again indicates the importance of an effective Slovene stemming algorithm.

7.3.3 Additional comparison of automatic stemming and manual right-hand truncation

In order to illustrate that these two types of search exhibit no significant performance difference in terms of retrieval effectiveness, some additional quantitative data can be presented. For example, Table 7.8 shows the frequency distribution of different documents, retrieved by automatic stemming and by manual right-hand truncation.

Number of different docs.	Freq
0	17
1	7
2	11
3	8
4	3
5	1
6	1
Total	48

Table 7.8: Frequency distribution of different documents, retrieved by automatic stemming and manual right-hand truncation

The most striking feature of Table 7.8 is the fact that processing of 17 queries (35.4%), using two different types of search (i.e., automatic stemming and manual right-hand truncation) resulted in the same top 10 documents. In addition, the ranking was entirely identical in 14 cases (29.2%).

In order to find out *why* there are some differences—although very little—between automatic stemming and manual right-hand truncation in the Slovene IR system, a simple *qualitative* investigation was carried out. Six queries were used as a sample for this type of evaluation, as illustrated in Table 7.9.

Query no.	Stemmed	Truncation
18	7	6
25	6	7
35	10	8
39	7	8
40	5	4
48	5	7

Table 7.9: Number of relevant documents retrieved by two types of search for six queries

While processing of queries 18, 35, 40 resulted in better performance of automatic stemming, processing of queries 25, 39, 48 resulted in a larger number of relevant documents retrieved by manual right-hand truncation. A detailed analysis of these six queries is presented below. First, queries where the stemming algorithm performed better will be analyzed, followed by the queries where the manual processing of terms performed better. The English translation of each query is given in square brackets.

QUERY 18:

Teorija in modeli samoupravnega javnega komuniciranja
[Theory and models of self-management mass communication]

A manual removal of endings, carried out by a trained intermediary, returned the following stems:

teor? model? samoupravn? javn? komunic?

It has been emphasized in Chapter 3 that the Slovene morphology is not only characterized by a large number of endings, but also by alterations which are carried out during word formation. If we consider the stem KOMUNIC* from the above list, its related words can also be KOMUNIKACIJA, KOMUNIKACIJAM, etc. Since the trained intermediary did not take into account these variations (-IC $\longrightarrow$ -IK), processing of the above query resulted in a smaller number of relevant documents (6) than the number obtained by the automatic conflation (7). The stemming algorithm has produced the stem KOMUNI* which was able to retrieve some other related terms from the relevant documents.

QUERY 35:

Citatna analiza (analiza citatov)
[Citation analysis]

The manual removal of suffixes, carried out by a trained intermediary resulted in the following list of stems:

citat? analiz?

The frequency of alterations which are conducted during word formation in Slovene is again demonstrated in this example. The related forms of the term CITAT include the following: CITIRANJE, CITIRANOST, etc. Consequently, the manual removal of suffixes was not capable of taking into account these alterations. On the other hand, the stemming algorithm can successfully process words such as CITATI, CITIRANJE, CITIRANOST, etc. and conflate them into CITAT, after using the recoding rule -IR $\longrightarrow$ -AT. As a consequence, the stemming algorithm retrieved a larger number of relevant documents (10) than did manual removal of suffixes (8).

QUERY 40:

Pomen mikrofilma in mikrofilmanja v knjižnicah in INDOK centrih
[Microfilm and microfilming in libraries and INDOC centres (centers; information and documentation services)]

The trained intermediary produced the following list of stems:

Pomen? mikrofilm? knjižni? INDOK? centr?

Whilst the trained intermediary conflated the word CENTRIH to CENTR*, the stemming algorithm produced the stem CENT*. This difference in conflation was the basic reason for the slightly better performance achieved by automatic conflation (5 relevant documents) than by manual right-hand truncation (4 relevant documents). Again, the stemming algorithm has not ignored the Slovene linguistic characteristics, i.e., the variants of the stem CENT* can be either words such as CENTER or terms such as CENTRI, CENTRIH, etc.

The above examples show that some of the very important linguistic rules of the Slovene language are included in the stemming algorithm. Since the professional intermediary was either not aware of these rules or they were not applicable to the manual right-hand truncation, the stemming algorithm was superior to the manual right-hand truncation in 12 queries. However, in eight cases the stemming algorithm was less successful in retrieving relevant documents within top 10 ranked documents. Some of the reasons why its performance was less effective are described below using a sample of three queries.

QUERY 25:

Izobraževanje (šolanje, vzgoja) knjižničarskih kadrov (bibliotekarjev, bibliotečnih kadrov), učni načrti in strokovni izpiti

[Training and education of librarians (library staff, personnel) – educational programmes (curriculums) and examination regulations]

From this query, the processing of two words had a crucial influence on the retrieval results. While the trained intermediary truncated terms ŠOLANJE and STROKOVNI to ŠOLAN* and STROKOV*, the stemming algorithm generated the stems ŠOL* and STROK* in order to retrieve more related words, i.e., to increase recall. However, the "strong" removal of suffixes resulted in only 6 relevant documents retrieved by the stemming algorithm; the manual truncation returned 7 relevant documents. The following are some examples of undesirable phrases which were brought in by the overstemming: ŠOLSKA KNJIŽNICA (SCHOOL LIBRARY), OSNOVNA ŠOLA (PRIMARY SCHOOL), ŠOLSKO LETO (ACADEMIC YEAR), and STROKOVNE KNJIŽNICE (SPECIAL LIBRARIES). These phrases are by no means related either to the phrase ŠOLANJE BIBLIOTEKARJEV (EDUCATION OF LIBRARIANS) or to the phrase STROKOVNI IZPITI (EXAMINATION REGULATIONS).

QUERY 39:

Analiza uporabnikov v knjižnicah
[User studies (surveys) in libraries]

The processing of this query, in particular the word UPORABNIKOV, shares similar characteristics to the previous query. The stemming algorithm has again, in order to increase recall, produced a string with fewer characters than one created by a trained intermediary, i.e., UPORAB* vs. UPORABN*. The consequence was the smaller number of relevant documents (7) than the number obtained by manual processing (8). Overstemming can be again illustrated by the retrieval of the following two phrases which are unrelated to the phrase ANALIZA UPORABNIKOV (USER STUDIES): UPORABA PROGRAMSKE OPREME (SOFTWARE USE), and UPORABA CITATOV (CITATION USE).

QUERY 48:

Centralni katalog – avtomatizacija

[Union catalogue – automation]

Manual right-hand truncation, carried out by the trained intermediary resulted in the list of following stems:

centraln? katalog? avtomat?

The employment of the stemming algorithm resulted in the following list of stems:

cent* katal* avtomat*

In this query, the stem CENT* was again produced by the stemming algorithm. Whilst its employment in query number 40 (as described above) resulted in better retrieval performance, the precision was reduced in this query. Moreover, the reduction of CENTRALNI to CENT* can be considered as an example of overstemming. The stem CENT* also retrieved phrases such as RAZISKOVALNI CENTER (RESEARCH CENTRE), INDOK CENTER (INDOC CENTRE), i.e., phrases which are by no means related to the phrase CENTRALNI KATALOG (UNION CATALOGUE). As a consequence, manual right-hand truncation retrieved a larger number of relevant documents (7) than automatic stemming (5).

A comparison of queries 40 and 48 illustrates how it is sometimes difficult to obtain a balance between over- and understemming in designing a stemming algorithm. However, as shown in previous sections, the Slovene stemming algorithm has produced good retrieval results in these experimental tests, i.e., performance differences between automatic stemming and manual right-hand truncation were not significant at all.

7.4 Conclusions

The main objective of *Experiment I* was to test whether automatic word conflation can be introduced into Slovene information retrieval systems with no average loss of

performance, thus allowing easier user access to the system. The experiment was also carried out to provide a basis for introducing other statistically-based techniques into a Slovene IR environment; so far, these techniques have been tested mainly on English language document collections.

The results of experimental testing have confirmed two main hypotheses of *Experiment I*:

- there is a significant performance difference between automatic word conflation and unstemmed processing of words in queries and documents;
- there is no significant performance difference between automatic stemming and manual right-hand truncation carried out by a trained intermediary.

It follows that one of the important components of an IR system, i.e., word conflation, can be automated in Slovene IR systems with no average loss of performance. If there are some signs that stemming does not make enough difference to retrieval of English documents (Harman, 1991, Keen, 1991b), the results above confirmed that devising an effective automatic means of stemming in a Slovene IR environment can significantly increase retrieval effectiveness.

Having obtained good performance results with the employment of automatic word conflation procedures, the next experiment, *Experiment II* (as described in Chapter 8) was carried out. Its main objective was to test the performance of statistically-based IR techniques in two different languages, i.e., Slovene and English. It was hoped that the results of this experiment could serve as an important contribution towards solving the problem of a multi-lingual approach to information retrieval.

Chapter 8

Multi-Lingual Approach to Document Retrieval

8.1 Introduction

If statistically-based techniques appear to work well for English (Willett, 1988a) there is no *a priori* reason why they should not work equally well for another language, in this case, Slovene. However, a comprehensive analysis, comparing English and Slovene retrieval performance, is required to confirm the correctness of the above assumption.

Therefore, the main problem to be investigated in Chapter 8 is contained in the following two questions:

1. Are statistically-based techniques applicable to Slovene IR systems?
2. Could statistically-based techniques provide a framework for developing a multi-lingual IR system?

Experiment II was designed and conducted in order to provide answers to the above two questions. The methodology and results of *Experiment II* are described in this chapter. First, the purpose and background for the experimental test are outlined. The following section on methodology consists of two sub-sections: an outline of the test environment;

and a description of the test procedures. The analysis and presentation of results serve as a basis for conclusions and suggestions for further work.

8.2 Purpose of the experiment

Starting from the above notions, the following three main hypotheses were introduced in *Experiment II*:

- *HYPOTHESIS 1*: Processing of the English documents and queries within a best-match environment will produce more or less identical hits to those retrieved from the Slovene document collection.
- *HYPOTHESIS 2*: There is no significant performance difference in retrieval effectiveness between Slovene queries and documents and their equivalents, translated in English.
- *HYPOTHESIS 3*: Processing of English and Slovene stems, using a string similarity measure, will produce a similar number of semantically related terms.

In order to carry out *Experiment II*, the methodology from *Experiment I* was employed. In other words, a need to control all variables of the experiment as far as possible (Robertson, 1981) dictated the application of a laboratory test. The decision to use this type of evaluation, rather than the operational test, meant that *Experiment II* was neither concerned with the performance of the complete IR system, nor were any of the user-oriented variables (e.g., human factors) taken into account.

Experimental testing required the following implementation of variables from *Experiment I*:

- enhancement of a test collection with documents and queries translated into English, and with additional relevance judgments;
- design and development of the English version of INSTRUCT for a PC;
- searching of documents in the English database;

- identification of semantically related terms from the English and Slovene dictionary component of the inverted file;
- evaluation of results.

Before implementation of the above variables is described in detail, a brief background for *Experiment II* is presented in the next section.

8.3 Background for the experiment

It has already been pointed out for *Experiment I* that, so far, there has been no published evidence of similar experiments carried out in Slovene IR systems. This comment can be extended to *Experiment II*. In other words, a comparison of the Slovene and English IR performances, based on the application of statistical IR techniques, can be described as a pioneering experiment in this area.

Moreover, the employment of statistically-based IR techniques in multi-lingual information systems is not only the first such experiment carried out in Slovenia, but also one of very few similar projects reported, so far, worldwide. One of the main reasons for the low interest in developing multi-lingual IR systems in general—not only by applying automatic methods—is the widespread use of the English language. Evidence shows that English is the international language for the communication of scientific and technical information. It is therefore not altogether surprising that large United States or British information services have not felt an urgent need to provide multi-lingual output.

Pressure for multi-lingual information systems has tended to come from countries or regions using major languages other than English. Countries within the EC—in particular Belgium—and other multi-lingual countries (Canada, USA, Yugoslavia, etc.) are doubtlessly among those in which end-users could benefit from a multi-lingual approach to document retrieval.

So far, the following two advantages have helped end-users in these countries to overcome the language barrier in document retrieval:

- a widespread knowledge of English in the scientific community;
- the ability of the professional intermediary to search databases using an English thesaurus or even uncontrolled terms.

In these countries, most information centres of sufficient sophistication to require access to large internationally available databases are likely to have at least one information intermediary with sufficient knowledge of English to carry out searches. Nevertheless, in spite of the dominant position of English there are undoubtedly areas where a multi-lingual approach to information systems is highly desirable. Apart from stressing a need for promoting information flows through wide multi-national involvement in global systems and networks, there is one particular factor that is gaining in importance. This factor—known as the *end-user*—seems to be increasing efforts by information researchers to develop techniques to overcome natural language difficulties. In other words, the provision of end-user searching facilities in document retrieval systems has been recognized as the only way to remove the barrier between the original source of a query and the query's answer.

For an IR system to be accepted by end-users, the following requirements must be met (Wolpert, 1983): speed, ease of use, high recall, relevance, and flexibility. In the field of language needs, the possibility for end-users to input a query and to receive answers in a language with which they have a high degree of familiarity, is doubtlessly one of the top requirements.

In the literature, the following four specific techniques for overcoming language problems in information transfer were considered:

- automatic translation of unprocessed text;
- automatic translation of preprocessed text;
- use of multi-lingual thesauri;
- switching languages.

Although *automatic translations* of both unprocessed and preprocessed text were not developed primarily for IR purposes, their potential use for abstracting services made them of particular importance in the late Seventies (Dubois, 1979). Automatic translation of *unprocessed text* is usually based on the following elements: a terminological dictionary (often a morpheme dictionary) to identify all words that are likely to be encountered; a transformation grammar, containing the rules of the source and the target languages together with translation rules; and a set of processing algorithms. The main problems with such procedures have proved to be ambiguities caused by homographs and prepositional reference. According to Dubois (1979), the application of these techniques might be useful only in highly specialized fields.

An attempt to eliminate the unreliability of the above technique was made by automatic translation of *preprocessed text*, using a limited number of syntactical forms or specified grammatical rules and a pre-established vocabulary. Again, this method can only be applied in fairly narrow areas, e.g., the study of certain industrial or chemical processes. This approach is implemented in TITUS, a system developed by the Institut Textile de France (Dubois, 1979).

The third technique is that of a *multi-lingual thesaurus*. Because of the absence of syntactical relationships, indexing in this case is straightforward. However, the largest problem seems to be the initial development of the thesaurus and its maintenance. Recent experience supports the view that building a multi-lingual thesaurus in several languages at the same time is a better method than translating an existing thesaurus. This is because the latter procedure generates homographs on a scale whose resolution leads to a weakening of thesaural structures in the target languages. This was confirmed experimentally by Sager *et al.* (1982) in building an integrated multi-lingual thesaurus for the social sciences. The authors emphasized the importance of planning thesauri multi-lingually from the start. Every addition of another language requires the total reassessment of all descriptors and it is therefore advisable to construct a thesaurus from the outset with a view to the various languages likely to be required in the future. It is interesting to note that such an approach was used in building the Serbo-Croat and English multi-lingual thesaurus for a database containing documents on law and

legislation (Martinović, 1985).

The use of *switching languages* permits the conversion of indexing terms used by a given centre into the terms used by a number of alternative centres (Dubois, 1979). With two or three indexing languages, tables of equivalents or appropriate conversion algorithms could link each term in one language with its conceptual equivalent(s) in another.

Although very little evidence can be found about the retrieval performance of the techniques described above, it seems that the majority of the IR systems interested in providing output in more than one language are heading towards the use of a multi-lingual thesaurus. However, according to Rolland-Thomas and Mercure (1989), there is still a very modest account of the ongoing research and writing on subject access in multi-lingual IR systems. The same comments can be made for multi-lingual OPACs, which have yet to appear.

8.3.1 Statistically-based techniques in multi-lingual IR systems

It has often been emphasized in previous chapters that statistically-based IR techniques allow greater end-user involvement in the searching process. It follows that these techniques could potentially be very attractive tools in designing multi-lingual IR systems.

The very little evidence about using statistically-based techniques in multi-lingual information systems—as revealed by a citation search in the LISA database—can therefore be considered as unexpected. Although one of the first experiments in this area was carried out by Salton (1969) almost three decades ago, very few similar reports can be traced afterwards in the literature.

Salton's experiment (1969) was a part of the SMART project. His starting point was the good experimental results that had been obtained by the employment of fully automatic text processing methods using relatively simple linguistic tools. These methods had been shown to be as effective for purposes of document indexing, classification, search, and retrieval as the more elaborate manual methods used in practice. Since all tests were carried out entirely with English language queries and documents, Salton

studied the extension of the SMART procedures to German language materials. A multi-lingual thesaurus was used for the analysis of documents and search requests, and tools were provided which made it possible to process English language documents against German queries and *vice versa*. The evaluation of these methods showed that the effectiveness of the mixed language processing was approximately equivalent to that of the standard process operating within a single language.

At the heart of Salton's experiment was a synonym dictionary, or thesaurus, which was used to recognize synonyms by replacing each word stem by one or more concept numbers; these concept numbers then served as content identifiers instead of the original word stems. A multi-lingual thesaurus was produced by manually translating into German an originally available English version. It is reported by Salton (1969) that the use of a thesaurus look-up process improves retrieval effectiveness by about 10% in both recall and precision.

The second known experiment in this area, carried out by Field (1977), investigated automatic multi-lingual indexing. French and English systems were compared during their automatic generation of thesaurus terms. Both systems produced successful and also equivalent results. On this basis, Field (1977) predicted a promising future for automatic indexing systems which employ multi-lingual thesaurus.

Despite this optimism as expressed by Salton (1969) and Field (1977) no similar experimental tests have been reported in the last decade. Moreover, Dubois (1979) claimed that these results were not clear enough to warrant serious consideration of these techniques. His conclusion was reinforced by the paucity of concrete applications of monolingual automatic indexing.

It has already been pointed out that the research work on the use of statistically-based IR techniques is now beginning to be reflected in operational systems, in particular in the English language-based IR environment. In addition, some of the important components of these techniques (e.g., automatic word conflation) have indicated the possibility for implementation of this approach in Slovene IR systems. Successful performance results obtained by the employment of these techniques in two different languages

have stimulated the consideration of automatic statistical methods for a multi-lingual information system. There is no doubt that end-users in Slovenia being surrounded by document collections written in different languages (i.e., Yugoslav languages and other major European languages) could benefit in the long term from the results of such an experiment.

8.4 Methodology

8.4.1 The test environment

The test environment in *Experiment II* consisted of the following components:

- a test collection;
- two versions (English and Slovene) of the information retrieval package INSTRUCT;
- a term expansion module, based on a measure of string similarity (trigrams) between a specified query term and each of the terms in the dictionary file;
- a best-match searching strategy.

All of the new components are briefly outlined in the next sections.

Test collection

A test collection consisting of the Slovene documents, queries, and relevance judgements was set up as a part of *Experiment I*. However, for purposes of the multi-lingual test, an English translation of the Slovene texts was required. Therefore the original test collection was extended with documents and queries translated into English. In addition, matching of English queries with the English documents required relevance judgments to be carried out by the same group of users as in *Experiment I*. A brief description of the English-based components of the test collection is presented below.

Documents. Most of the articles in both journals which were used for the Slovene document collection were also accompanied by abstracts, translated into English. Therefore, the process of building the English document collection was straightforward. The only problem was keyboarding which required three weeks work by the author. Because of the English automatic spell-checker, no particular additional checking of text was needed.

The following is an example of a Slovene unit (referred to as SLOV) and its English equivalent (referred to as ENGL):

SLOV:

Sodobni trendi v iskanju dokumentov

* Popovič, M.: Knjižnica, 34(1990)1/2, str. 9-31

Trend ekstenzivne rasti bibliografskih in tekstovnih podatkovnih zbirk ter razvoj na področju hardverske in softverske tehnologije postavljata vedno bolj v ospredje sodobne, nekonvencionalne sisteme za iskanje informacij. Predstavniki teh sistemov so že dokazali, da ne omogočajo le večjega števila relevantnih zadetkov, temveč tudi samostojno iskanje uporabnikov po podatkovnih zbirkah. V članku so prikazane naslednje sodobne tehnike iskanja informacij, ki temeljijo na uporabi statističnih metod: avtomatsko indeksiranje, iskanje podatkov po načelu optimalnega primerjanja in ponderiranje gesel. Hkrati so opisane tudi nekatere jezikovno odvisne procedure, ki so bile doslej razvite tudi že za slovenski jezik (npr. algoritem za avtomatsko zlivanje besed).

ENGL:

Current trends in document retrieval

* Popovič, M.: Knjižnica, 34(1990)1/2, pp. 9-31

The increasing use of bibliographic and text retrieval systems and developments in hardware and software technology will lead to a growing interest in advanced retrieval systems. These systems have already been shown to be able to retrieve larger amounts of relevant material than conventional systems and to replace trained intermediary by end-users unexperienced at the search process. In this article, the following statistically-based methods of advanced information retrieval are described: automatic indexing, best-match searching, term weighting. Language dependent procedures developed so far for Slovene are also briefly outlined (e.g., automatic word conflation algorithm).

The implementation of the English version of the INSTRUCT package required the processing of document collections. The English document collection was first filtered by the application of the list of English stop-words, followed by Porter's stemming

algorithm. After this complex processing was completed, the Slovene and English document collections exhibited similar characteristics in statistical terms. Table 8.1 shows some quantitative features of both collections, first after removal of stop-words, and secondly, after applying the stemming algorithms.

Quantitative characteristics	SLOV	ENGL
Number of word types (stop-words deleted)	8,602	4,756
Number of stem types	2,957	3,012

Table 8.1: Quantitative characteristics of the Slovene and English document collections

Table 8.1 points out the following main features of both document collections:

- The removal of stop-words preserves the difference in morphological complexity between Slovene and English; because of the morphological richness, the Slovene document collection contains a much larger number of original word types.
- The application of stemming algorithms reduces these differences to a minimum. As a result, the Slovene document collection is indexed by 2,957 stems, comparing to 3,012 stems for the English equivalent. It is important to note that the almost equivalent number of stems in both databases lends full credibility to *Experiment II*.
- A similar number of stems was achieved by 65.5% level of compression of the Slovene dictionary file, and by 36.7% level of compression of the English equivalent. Since larger vocabularies were used than those employed in Chapter 4, greater reductions in the size of the vocabularies were noted. It is interesting to see that this experiment confirmed Porter's conclusion (1980) that the employment of his algorithm results in about one third reduction in the size of the vocabulary file. His vocabulary consisted of 10,000 words; the suffix stripping process resulted in 6,370 distinct entries, achieving 36.3% level of compression.

A set of queries. In order to carry out *Experiment II*, an English translation of the Slovene queries was required. Table 8.2 illustrates the main comparative characteristics of the Slovene set of queries and the English equivalent after translation was complete.

Quantitative characteristics (before stopwording)	SLOV	ENGL
1. Number of queries	48	48
2. Total number of terms in a set of queries	370	399
3. Average number of terms per query	7.7	8.3
4. Maximum number of terms per query	19	21
5. Minimum number of terms per query	2	2

Table 8.2: The main quantitative characteristics of the Slovene and English sets of queries

Table 8.2 indicates minor differences in the frequency distribution of terms between the Slovene set of queries and its English counterpart. The latter displays a slightly larger total number of words (i.e., 399 terms in the English set compared to 370 terms in the Slovene set). Consequently, a slight difference in the average number of terms per query was also noted (8.3 terms per query in the English set; 7.7 terms per query in the Slovene set). These differences have been generated mainly by the following:

- the translation service;
- differences between Slovene and English grammar.

Although an attempt was made to translate the queries from Slovene to English on the basis of a precise consistency, some of the queries required a broader accommodation of the English terminology. An example can be found in the Slovene phrase *INDOK centre* which is usually translated as *information and documentation centre* and very rarely as *INDOC centre*. The following is an example of the Slovene query and its translation into English:

QUERY 6:

SLOV: Specializirani INDOK centri v Sloveniji in na Hrvaškem.

ENGL: Specialized INDOC centres (information and documentation services) in Slovenia and Croatia.

The above example illustrates that the English query is characterized by a larger number of terms (11) than the Slovene query (8). The second reason for quantitative differences between English and Slovene query sets can be found in grammar. The English language is characterized by the frequent use of articles (i.e., A, AN, THE) which are not used in the Slovene language at all. This is demonstrated by the following query.

QUERY 23:

SLOV: Obvezni izvod in slovenska nacionalna bibliografija

ENGL: The deposit copy and the Slovene National Bibliography

The English query again has a greater number of terms (8) than its Slovene counterpart (6). Since terms such as A, AN, THE are included in the list of stop-words used by English IR systems, it is interesting to compare quantitative characteristics of both sets of queries after removal of stop-words. This is illustrated in Table 8.3.

Quantitative characteristics (after stopwording)	SLOV	ENGL
1. Number of queries	48	48
2. Total number of terms in a set of queries	293	292
3. Average number of terms per query	6.1	6.1
4. Maximum number of terms per query	15	13
5. Minimum number of terms per query	2	2

Table 8.3: The main characteristics of the Slovene and English sets of queries after deletion of stop-words

The removal of stop-words from both sets of queries resulted in almost identical quantitative characteristics (i.e., the English set having in total 292 words, comparing to 293 words in the Slovene set). This is, of course, an important argument for the correctness of the multi-lingual experiment. The validity of the project was further confirmed by the application of the stemming algorithms to the terms in the two sets of queries, as demonstrated in Table 8.4.

Quantitative characteristics	SLOV	ENGL
Number of word types (stop-words deleted)	224	159
Number of stem types	148	144

Table 8.4: Quantitative characteristics of the Slovene and English query sets before and after the employment of automatic stemming

Table 8.4 shows again—as also demonstrated in Table 8.1—that two completely different stemming algorithms (i.e., Slovene and Porter's) are able to produce almost identical numbers of stem types. This is, of course, of crucial importance for the statistical rigour of the experiment. A list of queries translated into English can be found in Appendix D, and then compared with the list of Slovene queries as presented in Appendix B.

Relevance assessments. Relevance judgments in *Experiment II* were carried out in the following manner:

- relevance assessments of a set of retrieved documents were made by the same group of users as in *Experiment I*;
- users were given the same instructions about carrying out relevance assessments as in *Experiment I*;

- users judged documents for their relevance on the basis of the information contained in the title and abstract of each retrieved document;
- the set of documents to be judged for relevance consisted of titles from best-match searching, using the ranked-output cutoff procedure; again, only the first ten retrieved documents (i.e., cutoff point was 10) were used for a further analysis;
- these documents were added to the pool developed in *Experiment I*;
- for the purpose of conducting a quantitative comparison, an additional pool was created containing documents retrieved by best-match search only—using automatic stemming—from both document collections; thus, both pools were a mixture of Slovene and English documents;

It has to be emphasized that all requesters were fluent in English. This means that their consistency in relevance judgments from *Experiment I* was more or less maintained.

Apart from judging retrieved English documents for their relevance, a parallel process was carried out as part of *Experiment II*. This method is known as a *query expansion* experiment, based on a string similarity measure. The main objective of this test was to retrieve and identify terms which are semantically related to a query term. A similar number of related retrieved terms from the Slovene and English dictionary should indicate that this statistically-based method could play an important role within a multi-lingual IR environment.

English version of INSTRUCT

The English variation of the INSTRUCT package was created on the basis of the Slovene version. This means that the new English variation of INSTRUCT required only language-dependent procedures (mainly deletion of stop-words and automatic conflation) to be modified. Again, the TURBO PASCAL 5.5. programming language was used.

8.4.2 The test procedures

Collection of data

To obtain the data required for the comparative evaluation, two main procedures were employed on the English document collection: best-match searching and identification of related stems. The best-match search required the application of the following steps:

- searching across a document collection, using the set of 48 English queries;
- retrieving a pool of document records that were then judged relevant or non-relevant by requesters;
- comparative evaluation of the results with the Slovene data from *Experiment I*.

A query expansion experiment was employed separately in the Slovene and English versions of the INSTRUCT package. Its application involved the following steps:

- searching in the dictionary component of the inverted file for stems which are similar to the selected query stems;
- retrieving, for each query stem, a pool of the 10 most similar stems (i.e., the stems having the largest number of trigrams in common with the selected query stem);
- assessments of the retrieved stems for their semantical relationship with the query stem;
- comparative evaluation of the resulting English and Slovene data.

Methods for the analysis of data

The employment of either best-match searching procedures or a query expansion module also affected the use of methods for the analysis of data.

Results provided by the best-match search in Slovene and English database were analyzed by the following methods:

- a simple quantitative comparison;
- a measurement of the retrieval effectiveness;
- a failure analysis.

The idea of using a simple quantitative comparison between Slovene and English retrieval results (i.e., the number of identical hits) was very direct: to obtain preliminary results either on differences or similarities between two systems. This means that items retrieved from the English database were compared to documents retrieved from the Slovene collection. It was expected that the frequency distribution of the identical hits (and, in particular, identical relevant documents) should give indicators for using other methods for the comparative evaluation. This assumption was confirmed during the course of the experiment.

The second method employed in *Experiment II* consisted of the retrieval effectiveness measures, i.e., recall and precision. In addition, a statistical significance test—the sign test—was performed on the difference between the Slovene and English systems.

A third method which was also used in this experiment is known as failure analysis. Its main objective was in providing answers to why certain relevant documents were not retrieved (Bawden, 1990). This type of analysis was shown to be extremely useful, in particular in demonstrating the ability to explain the differences between Slovene and English retrieval performance.

The results obtained by the multi-lingual query expansion experiment were analyzed by the following two methods:

- a simple quantitative comparison;
- a failure analysis.

A simple quantitative comparison was employed to provide an answer on the similarities between English and Slovene. These similarities were calculated on the basis of retrieving a certain percentage of semantically related terms in each system. Since it

was difficult to expect that both systems would provide identical numbers of related stems per term, a failure analysis was also employed.

8.5 Analysis of results

8.5.1 Multi-lingual experiment, using best-match searching facility

As stated above, the performances of the Slovene and English versions of INSTRUCT—from now on referred to as SLOV and ENGL, respectively—were first compared and analyzed within a nearest neighbour searching module. The results, analyzed by the employment of different methods, are described in the next sections.

A simple quantitative comparison

A pool of Slovene retrieved relevant documents was defined as a target for the comparative component in *Experiment II*. In other words, this simple analysis considered only those retrieved English documents which were *identical* to the Slovene documents. This means that non-identical English retrieved items were analyzed later within the context of measuring the retrieval effectiveness of both systems.

After the INSTRUCT package had been applied to the English document collection, the following quantitative similarity—expressed in terms of number of identical documents—with the Slovene hits was determined. The processing of 48 English queries resulted in 243 hits (50.6%) which were also found in the Slovene list. In other words, the English version of INSTRUCT retrieved 237 documents (49.4%) which were not found by the Slovene set of queries (since 480 documents were retrieved in all).

The relatively low similarity between Slovene and English lists of retrieved documents has been partly improved by comparing only relevant documents from both listings. The English version of INSTRUCT was capable of retrieving 189 relevant documents which were also present in the Slovene list, consisting of 302 relevant items. This percentage of similarity was therefore 62.6%. On the other hand, only 54 identi-

cal non-relevant English documents were found in the Slovene listing, containing 178 non-relevant items (30.3%). Results of this simple comparison are summarized in Table 8.5.

SLOV: 480 hits ENGL: 243 identical hits (50.6%)
SLOV: 302 relevants ENGL: 189 identical relevants (62.6%)
SLOV: 178 non-relevants ENGL: 54 identical non-relevants (30.3%)

Table 8.5: Percentage of the English retrieved documents which are identical to the documents in the Slovene list

Table 8.5 demonstrates that the identical documents comprise a much larger percentage of relevant documents (62.6%) that non-relevant hits (30.3%). However, the proportion of retrieved failures (i.e., relevant documents retrieved only by the Slovene version of INSTRUCT, and not by the English one)—37.4%—begin to raise some initial doubts about using statistically-based IR techniques in multi-lingual information systems.

Table 8.6 additionally illustrates the level of similarity between the English and Slovene capability in retrieving identical relevant documents. It is evident that only 8 queries (16.7%) resulted in a list of entirely identical relevant documents. In most examples, the difference between English and Slovene lists of relevant documents was expressed either as 1 document (10 queries, or 20.8%) or as 2 documents (13 queries, or 27.1%). In total, 31 queries (64.6%) produced results where the difference between the Slovene and English listings was expressed either by none, one or two relevant documents.

Number of different documents	Freq
0	8
1	10
2	13
3	3
4	6
5	3
6	3
7	1
8	1
Total	48

Table 8.6: Frequency distribution of different relevant documents, retrieved by the English and Slovene versions of INSTRUCT.

However, 17 queries (35.4%) were still retrieving three or more relevant documents which were not presented in either the English or Slovene listing. This percentage again raised the question of whether simple statistically-based techniques—as implemented in INSTRUCT—are applicable as a straightforward method in designing multi-lingual IR systems. In order to be able to answer this question more data on the performance of both systems was required.

Recall and precision as measures of retrieval effectiveness

A measurement of the retrieval effectiveness was based on information from the pool of retrieved documents in *Experiment I*. The comparative analysis required this pool to be incremented with documents retrieved from the English database. This meant that—having applied a cutoff of 10—each query could be potentially represented by 40 distinct retrieved documents.

It has to be noted again that a relative recall was calculated, based on the substitute

lists of total relevant items in the collection (i.e., a pool of retrieved documents). On the basis of comparison of relevant items from the Slovene pool with the top 10 documents on the English list the absolute figures were obtained for each query, and then recall values calculated.

Table 8.7 clearly shows that applying the INSTRUCT package for searching document in the Slovene collection produces better results than its employment in retrieving documents in the English database. In other words, whilst the Slovene version of INSTRUCT retrieved 302 relevant documents (achieving a recall of 69%), its English equivalent produced a list with only 248 relevant documents (achieving a recall of 56%). This means that retrieval of items from the English document collection reported a total number of 54 retrieved failures.

The precision figures were also extracted from Table 8.7, using the mean number of relevant documents, multiplied by 10. Since the mean number of relevant documents in the Slovene pool was equal to 6.3 documents, the value of precision was 63%. On the other hand, the mean number of relevant hits in the English pool was equal to 5.2, resulting in a precision of 52%. The average recall and precision values of the two systems are summarized in Table 8.8.

Although the English version of INSTRUCT retrieved some new relevant documents (its contribution to the *total* pool was 36 new relevant items, i.e., 8.2%), results in Table 8.7 and in Table 8.8 indicate a quite large performance difference between the two systems. In other words, the Slovene version of INSTRUCT was much more successful in retrieving relevant documents. In order to find out whether this difference could be regarded as a statistically significant difference, the sign test was carried out.

The sign test. The existence of the significant performance difference was tested by the employment of the following hypotheses:

- H_0 : the application of the Slovene and English versions of INSTRUCT results in the retrieval of a similar percentage of relevant documents;

Query	SLOV		ENGL		Pooled ret/rel	Total distinct retrieved
	rel	%	rel	%	no. docs	no. docs
1	6	100	4	67	6	23
2	9	90	8	80	10	18
3	9	90	9	90	10	20
4	7	87	7	87	8	17
5	8	80	7	70	10	21
6	8	89	7	78	9	21
7	10	56	8	44	18	23
8	3	50	5	83	6	21
9	7	54	5	38	13	22
10	7	70	5	50	10	21
11	9	56	7	44	16	21
12	10	63	5	31	16	24
13	8	89	8	89	9	16
14	9	75	9	75	12	17
15	9	69	7	54	13	15
16	5	83	4	67	6	21
17	7	100	5	71	7	21
18	7	100	4	57	7	18
19	6	38	3	19	16	27
20	10	50	6	30	20	25
21	10	77	8	62	13	22
22	6	67	6	67	9	12
23	6	75	7	88	8	18
24	3	75	3	75	4	28
25	6	43	7	50	14	23
26	7	78	6	67	9	20
27	8	73	5	45	11	18
28	7	54	8	62	13	19
29	5	56	4	44	9	24
30	1	50	2	100	2	20
31	5	56	2	22	9	28
32	3	60	3	60	5	19
33	5	100	5	100	5	14
34	4	67	2	33	6	20
35	10	77	8	62	13	20
36	9	90	4	40	10	20
37	3	75	2	50	4	27
38	4	57	5	71	7	23
39	7	70	4	40	10	25
40	5	71	3	43	7	23
41	3	37	2	25	8	26
42	2	67	2	67	3	23
43	6	100	6	100	6	23
44	5	62	5	62	8	21
45	6	100	5	83	6	20
46	2	50	2	50	4	26
47	5	71	2	29	7	22
48	5	71	7	100	7	23
Total	302	69	248	56	439	1,029

Table 8.7: Number of relevant retrieved documents and aggregate recall for the Slovene and English versions of INSTRUCT (cutoff 10)

Retrieval effectiveness	SLOV	ENGL
RECALL	69	56
PRECISION	63	52

Table 8.8: The average recall and precision values of the Slovene and English versions of INSTRUCT (cutoff 10)

- H_1 : the employment of the Slovene package results in a larger number of relevant documents than use of the English system.

The significance level (α) was defined as 0.05, the number of couples (N) was equal to the number of queries, i.e., $N = 48$, and the rejection region was one-tailed.

Table 8.9 shows that the Slovene package was more successful in 29 observations, and less successful than the English system in only 7 observations. The number of tied cases was 12.

Sign	Freq
+	29
−	7
0	12
Total	48

Table 8.9: Frequency distribution of the direction of differences between the Slovene and English versions of INSTRUCT

All tied cases were dropped from further analysis; consequently N was reduced to 36. Since N was still larger than 35, a method for large samples was applied (see Siegel

and Castellan, 1988), resulting in a value of $z = 3.5$. Reference to Table A in Siegel and Castellan (1988) reveals that the probability $z >= 3.5$ when H_0 is true is 0.00023.

Since 0.00023 is smaller than $\alpha = 0.05$, the decision was made to reject the null hypothesis in favour of the alternative hypothesis. Thus, the sign test proved that there is a significant performance difference between the Slovene and English systems. Or, in other words, the Slovene version of INSTRUCT is able to produce significantly better results than its English counterpart.

These results, i.e., a low number of identical retrieved documents and a significant performance difference between the two systems was unexpected. As a result, the two main hypothesis of *Experiment II* were rejected by this experiment. In general, the experimental results gave the following indications:

- the more than satisfactory performance of the Slovene version of INSTRUCT indicates that statistically-based techniques can be introduced in Slovene IR systems without any hesitation;
- the significant performance difference between the Slovene and English versions of INSTRUCT has cast a serious shadow on the use of statistically-based techniques (as implemented in INSTRUCT) as a suitable tool in developing multi-lingual information systems.

This shadow was enlarged further after the so-called *best-match pool* was created, consisting only of Slovene and English documents, retrieved by using automatic word conflation within a best-match context. All relevant documents from this pool were analyzed in order to find out their origin, i.e., whether they had been retrieved by the Slovene version of INSTRUCT or by its English equivalent. Table 8.10 illustrates the frequency distribution of relevant documents according to their source.

Table 8.10 shows again that the Slovene system was superior to the English system. While the Slovene version of INSTRUCT retrieved 113 relevant documents (31.3%) not found by the English version, the latter reported 59 relevant items (16.3%) not retrieved by the Slovene system. In addition, the percentage of identical relevant documents

Source	Freq	%
Relevants retrieved by both SLOV and ENGL	189	52.4
Relevants retrieved only by SLOV	113	31.3
Relevants retrieved only by ENGL	59	16.3
Total	361	100.0

Table 8.10: Frequency distribution of retrieved relevant documents according to their source

retrieved by both systems was relatively low, i.e., 52.4%.

In order to find reasons for the above expressed differences between the English and Slovene systems, and on this basis to suggest further solutions, a failure analysis was carried out.

Failure analysis

The employment of the failure analysis resulted in the identification of several possible causes of the Slovene and English retrieved failures. The discussion in this section will focuss on the following:

- the quality of the translation service;
- problems associated with the natural language processing of queries and documents;
- performance differences between the Slovene and English (i.e., Porter's) stemming algorithms.

Having obtained a large number of retrieved failures in the English version of INSTRUCT (i.e., 113 documents, or 31.3%) the first cause of failure which came to mind

was poor translation service. It is exactly this issue which was considered first. In addition, despite many advantages of the natural language query input and automatic indexing, there is one main problem which could also contribute to failures in retrieval. This problem is the question of how to cope with synonyms and other related terms without having, for example, a look-up thesaurus or other useful tools as a part of the system.

Finally, automatic word conflation—as a language-dependent procedure—was also considered in detail as a part of the failure analysis. Since both algorithms, i.e., the Slovene algorithm and Porter's algorithm, were developed independently from each other, and are based on different principles, they can also potentially seriously affect retrieval performance. For example, whilst Porter's algorithm correctly treats a certain term, the Slovene algorithm can produce an overstemming of the same term, and *vice versa*. Consequently, differences in stem weights can appear, leading to inconsistencies in ranking of documents.

To take into account possible causes of retrieved failures—as enumerated above—the analysis was carried out as follows:

1. processing of all queries from the Slovene and English sets;
2. deletion of stop-words and application of both stemming algorithms; on this basis a list of query stems was obtained, illustrating the following details:
   (a) the number of documents in which a particular stem occurred;
   (b) the weight calculated for this stem;
3. employment of best-match searching, resulting in two lists (Slovene and English) of the first ten retrieved documents for each query;
4. application of failure analysis to find causes for retrieved failures;
5. extension of the ranked cutoff point from 10 to 20 in order to check whether some of the missing documents are within the first 20 retrieved items;

6. analysis of the frequency distribution of the causes of retrieved failures in order to find out which factor affected the most performance differences between the English and Slovene systems.

After the results of the detailed, time-consuming, failure analysis had been obtained, it became possible to produce Table 8.11 which illustrates the contributions of these factors to retrieved failures.

Cause of failure	SLOV	ENGL
Poor translation	12	12
"Uncontrolled" occurrence of synonyms	35	63
Stemming algorithm	12	38
Total	59	113

Table 8.11: Frequency distribution of the main causes of retrieved failures in the English and Slovene systems

A detailed analysis of these main factors—the most important being the occurrence of synonyms and other related terms in the text—is given below.

Translation service. A *poor translation* was interpreted as the cause of retrieved failures for the following:

- incorrectly written words (e.g., SPLOŠNO IZOBRAŽEVALNE which should be written as a single word);
- improper translation (e.g. YUGOSLAVIA – NAŠA DRŽAVA; NATIONAL AND UNIVERSITY LIBRARY OF SLOVENIA – NARODNA IN UNIVERZITETNA KNJIŽNICA V LJUBLJANI)

SLOV	ENGL
—	INDOC
naša država	Yugoslavia
—	UNISIST
projekt	—
—	Slovene
Pionirska	Pionirska
splošno izobraževalne	public
Ljubljana	—
—	automatic
Ljubljana	Slovenia

Table 8.12: Examples of poor translation service

As shown in Table 8.11, the poor translation service resulted in 12 Slovene relevant documents (3.3%) not being retrieved; exactly the same number of retrieved failures was also reported for English documents. Table 8.12 illustrates some examples of bad translation ("—" indicates that the term was not translated at all).

The effect of poor translation (i.e., not translated terms, unsuitably translated words, etc.) is, of course, evident in the final ranking of retrieved documents. This can be illustrated with the employment of two queries:

QUERY 36:

Karkoli o Narodni in univerzitetni knjižnici (NUK) v Ljubljani.

Anything about the National and University Library in Ljubljana.

The processing of this query (NUK is the abbreviation for the National and University Library) resulted in the following list of stems (numbers in parentheses indicate document frequency, numbers in the square brackets show the weight of a particular stem).

SLOV		ENGL	
NAROD	(27) [3.8717]	NATION	(76) [2.7284]
UNIVER	(85) [2.5952]	UNIVERS	(118) [2.1832]
KNJIŽ	(240) [1.0953]	LIBRARI	(247) [1.0397]
LJUBLJAN	(43) [3.3722]	LJUBLJANA	(46) [3.2982]
NUK	(11) [4.8267]		

Processing of the Slovene query resulted in 9 relevant documents, and its English equivalent retrieved only 4 relevant items. The following example shows why an English document became a retrieved failure.

2/113
Referalna literatura — instrument informatorja (Ob oblikovanju kataloga referalne literature v NUK)

* Jakac-Bizjak, V.: Knjižnica, 29(1985)1, str. 74-79

Prispevek prinaša nekaj misli o referalni literaturi in o informacijski službi kot je Narodna in univerzitetna knjižnica v *Ljubljani.* ...

28/113
Reference literature — the instrument of the information librarian (The promotion of a new reference literature catalogue in the National and University Library of *Slovenia*)

* Jakac-Bizjak, V.: Knjižnica, 29(1985)1, pp. 74-79

This contribution represents some new approaches to the reference literature and information service in the libraries such as National and University Library of *Slovenia.* ...

The above example illustrates the improper translation of the word LJUBLJANA to SLOVENIA as emphasized by italics in the text. Since the term SLOVENIA did not appear in the query, the English document was ranked at the 28th position. In contrary, its Slovene equivalent achieved a high, 2nd place.

QUERY 23:

Obvezni izvod in slovenska nacionalna bibliografija.

The deposit copy and the Slovene National Bibliography.

Again, the processing of this query produced the following list of stems:

SLOV		ENGL	
OBVEZ	(9) [5.0073]	DEPOSIT	(7) [5.2627]
IZVOD	(3) [6.1180]	COPI	(10) [4.9000]
SLOVEN	(74) [2.7597]	SLOVENE	(41) [3.4242]
NACIJ	(43) [3.3722]	NATION	(76) [2.7284]
BIBLIOGRAF	(49) [3.2285]	BIBLIOGRAPHI	(26) [3.9115]

While the English version of INSTRUCT retrieved 7 relevant documents, its Slovene equivalent found 6 relevant items. In addition, the Slovene version reported 2 retrieved failures. The main cause of failure was again a poor translation service, this time in Slovene texts. One of the Slovene retrieved failures is presented below.

> 8/94
> Description of monograph publications in Slovene *National* Bibliography and National Bibliography of Yugoslavia
> * Zeni, J.: Knjižnica, 28(1984)1/2, pp. 16-34
> Slovene *National* Bibliography and National Bibliography of Yugoslavia of 1975 have already been written according to ISBD(M), ...
>
> 18/94
> Opis zaključenih publikacij v Slovenski bibliografiji in Bibliografiji Jugoslavije
> * Zeni, J.: Knjižnica, 28(1984)1/2, str. 16-34
> Slovenska bibliografija in Bibliografija Jugoslavije sta v popisih za leto 1975 že prešli na ISBD(M). ...

The above example shows that the absence of the word NACIONALNA in the Slovene document caused this document to be ranked as 18th. Since the word NATIONAL was present in the English document, its ranking was much higher, i.e., this document was among the first ten retrieved items. The second Slovene retrieved failure was again generated by the omission of the word NACIONALNA from the text.

These examples demonstrate that the translation of documents from Slovene to English could be carried out by a better service. This comment is, in particular, a criticism of the journal *Knjižnica* whose translation service should achieve significant improvements in the future.

Natural language processing of queries and documents. The main barrier towards a more effective multi-lingual approach was caused—as evident from Table 8.11—by *natural language*. A frequent occurrence of synonyms and related terms is a

feature of natural language which cannot be controlled during translation of texts. In addition, natural language processing in INSTRUCT is based on weighted *single* terms. In other words, as stated by Salton (1988) "...it is obviously not the case that the full text content is easily representable by single term sets".

There is no doubt that synonyms and other related terms whose appearance within INSTRUCT was not controlled, contributed the most to the significant performance difference between the Slovene and English systems. The following is a list presenting some examples of Slovene and English synonyms and related terms from both document collections:

SLOV:
Velika Britanija — Great Britain, United Kingdom
iskanje — searching, retrieval
poizvedba — query, question
javen — public, mass
eksponat — object, item

ENGL:
index — indeks, ključna beseda
software — softver, program, programska oprema
analysis — analiza, obdelava
staff — delavci, osebje, kader
training — usposabljanje, vzgoja
nation — nacija, narod
Yugoslavia — Jugoslavija, SFRJ
data base — podatkovna zbirka, baza podatkov, datoteka
user — uporabnik, bralec

These are only some of the examples which illustrate problems when it comes to the translation of a certain term. As shown in Table 8.11, the existence of synonyms and related terms caused 35 Slovene retrieved failures (9.7%). This percentage was even larger in the English system which reported 63 retrieved failures (17.4%). The examples below illustrate the effect of natural language on retrieval performance.

QUERY 2:

Knjižnice in informacijski centri v Veliki Britaniji in Indiji.

Libraries and information centres (centers) in Great Britain and India.

The processing of this query produced the following list of stems:

SLOV		ENGL	
KNJIŽ	(240) [1.0953]	LIBRARI	(247) [1.0397]
INFOR	(261) [0.9285]	INFORM	(261) [0.9285]
CENT	(101) [2.3838]	CENTR	(32) [3.6912]
		CENTER	(50) [3.2061]
VELIK	(47) [3.2745]	GREAT	(25) [3.9528]
BRITAN	(7) [5.2627]	BRITAIN	(4) [5.8283]
INDIJ	(6) [5.4188]	INDIA	(6) [5.4188]

While the Slovene version of INSTRUCT retrieved 9 relevant documents, the English processing of this query resulted in 8 relevant documents. One retrieved failure was caused by the occurrence of a related term. In other words, the phrase VELIKA BRITANIJA was translated in all documents to GREAT BRITAIN, apart from one item where the term UNITED KINGDOM was used.

1/154
EADI bibliotečno-dokumentacijsko-informacijska družina se je ponovno sestala
* Potočnik-Kovše, T.: Knjižnica, 31(1987)1, str. 89-99
Prispevek je kratko poročilo o EADI seminarju v Brightonu (*Velika Britanija*), septembra 1986, ...

40/154
EADI library-documentation-information family has been afresh brought together
* Potočnik-Kovše, T.: Knjižnica, 31(1987)1, pp. 89-99
The contribution is a brief report on EADI meeting in Brighton (*United Kingdom*), September, 1986, ...

Since UNITED KINGDOM was not included in the query, its occurrence in the document (as a substitute for GREAT BRITAIN) caused a significant decline of this document in the output list. A similar situation was reported also for one Slovene document as illustrated below.

QUERY 41:

Specializirane baze podatkov (podatkovne zbirke) v Jugoslaviji.

Specialized databases (data bases, collections) in Yugoslavia.

The following list of stems was obtained:

SLOV		ENGL	
SPECIAL	(42) [3.3979]	SPECIAL	(101) [2.3838]
PODAT	(106) [2.3230]	DATA	(106) [2.3230]
BAZ	(22) [4.0869]	BASE	(65) [2.9101]
		DATABAS	(16) [4.4177]
ZBIR	(51) [3.1841]	COLLECT	(47) [3.2745]
JUGOSL	(66) [2.8926]	YUGOSLAVIA	(52) [3.1624]

Although the Slovene version of INSTRUCT retrieved more relevant documents than its English counterpart (3 *vs* 2) there was still 1 retrieved failure reported in the Slovene list. This document is analyzed below.

> 3/145
> A short description of the development of Slovene library information system with respect to building industry
> * Verbič, D., Perc-Kovačič, C., Majcen-Čučnik, N.: Knjižnica, 30(1986)3/4, pp. 81-90
> ... General standpoints of the *Yugoslav* library information system derive from the basic principles of UNISIST. ...

> 21/145
> Kratek opis razvoja knjižnično-informacijskega sistema v SR Sloveniji (KIS) s poudarkom na graditeljstvu
> * Verbič, D., Perc-Kovačič, C., Majcen-Čučnik, N.: Knjižnica, 30(1986)3/4, str. 81-90
> ... Širša izhodišča KIS v *SFRJ* izhajajo iz osnovnih načel razvoja svetovnega sistema znanstvenih informacij — UNISIST. ...

The absence of the word SFRJ (a synonym for JUGOSLAVIJA) from the query caused the above Slovene document—ranking 3rd in the English list—to be dropped to 21st place. Since a ranked cutoff was defined at 10, this document was not presented in the pool of retrieved items; it was, consequently, considered as a retrieved failure.

The above examples have demonstrated that the main problem derives from the question of how to improve the content analysis of a multi-lingual text. There is no doubt that automatic indexing techniques (assignment of weights to the single index terms) cannot entirely represent the full text content in two languages. One of the best known techniques which can cope with synonyms and other related terms is known as *controlled indexing*. This technique provides a representation of a wide variety of related

terms and descriptors by a single standard term or phrase. Such an indexing process is usually controlled by a thesaurus that contains classifications of similar words, and it is useful in broadening the indexing vocabulary by supplying synonyms and other related words.

There is no doubt that the implementation of a thesaurus look-up process in INSTRUCT would improve its multi-lingual retrieval effectiveness. A multi-lingual thesaurus would contribute to the recognition of synonyms and other related words by replacing the original word stems with the corresponding thesaurus categories. Unfortunately, the problem is that thesauri constructed from particular document collections are not easily applied to new situations and new collections. Hence a given thesaurus often provides improvements that are valid only locally and under special circumstances (Salton, 1988).

This is a very important point when considering possible techniques for improving a statistically-based approach in a multi-lingual IR environment. Any refinements based on the application of a multi-lingual thesaurus are bound to the specific document collections and subject areas.

Stemming algorithm. It is pointed out by Porter (1980) that the employment of suffix stripping usually results in a performance significantly less than 100%. The main function of the stemming algorithm design—in Porter's words—is to keep the balance between the number of stemming rules and the efficiency of processing.

Results in Table 8.11 are in correlation with Porter's statement. It can be seen that his algorithm produced 38 retrieved failures. In other words, 10.5% of missing relevant English documents were due to the "errors" in suffix stripping. This percentage was much lower for Slovene documents where only 12 missing relevant documents (3.3%) were reported. It has to be noted immediately that the successful performance of the Slovene stemming algorithm probably derives from the fact that this stemming algorithm (in particular, the list of suffixes) was produced using a text corpus from the librarianship and information science field. As it is known, the Slovene test collection in

Experiment II covers exactly the same area. The employment of a document collection from any other field (e.g., engineering) would probably increase the number of retrieved failures also for the Slovene stemming algorithm.

However, it is interesting to note that—despite the presence of both types of stemming errors—Porter's algorithm has been in particular affected by understemming. A similar conclusion was also reported by Lennon *et al.* (1981). Some examples of both types of errors, as encountered in the queries and documents from the test collection in *Experiment II*, are presented below.

1. understemming:

   (a) SLOV: muzeol – muzej; avtomat – avtomatiz; baz – bazič; domoznan – domoznanst; softv – softver; strok – strokov;

   (b) ENGL: Britain – British; Slovene – Slovenia; Yugoslav – Yugoslavia; scientif – scientist – scienc; librarianship – librari; analysi – analyt – analyz; museum – museologi; bibliograph – bibliographi; comput – computer; method – methodolog – methodologi;

2. overstemming:

   (a) SLOV: REFER – referalni, referat; KNJIŽ – knjižnica, knjiga; CENT – centralni, center; GRAD – gradnja, gradivo;

   (b) ENGL: PUBLIC – public, publications; RELAT – relations, related; IDENT – identity, identical; UNIVERS – university, universal; COMMUN – communication, community.

Both types of errors can seriously affect multi-lingual retrieval of documents. While understemming results in many relevant documents not being retrieved at all, overstemming generates retrieval of non-relevant items. Both errors are in particular evident by the assignments of weights to word stems. Consider the following example. The application of Porter's algorithm to terms UNIVERSAL and UNIVERSITY resulted in the stem UNIVERS–. This stem occurred in 118 documents (its weight was equal to

2.185). On the other hand, the Slovene stemming algorithm conflated UNIVERZALNA to UNIVERZAL (occurring in 12 documents, with weight 4.714) and UNIVERZA to UNIVER- (occurring in 85 documents, with weight 2.596). These differences, of course, produce different rankings of the Slovene and English documents, and—at the end—different retrieval performance.

This is illustrated by the employment of two queries from the test collections. Since both queries are representative of very short statements, one would expect identical lists of Slovene and English retrieved documents: in fact, these two queries will demonstrate how the employment of two different stemming algorithms can seriously affect performance results.

QUERY 12:

Splošnoizobraževalno knjižničarstvo v Ljubljani.

Public librarianship in Ljubljana.

The processing of this query resulted in the following list of stems:

SLOV		ENGL	
SPLOŠNOIZOBRAZ	(36) [3.5649]	PUBLIC	(132) [2.0361]
KNJIŽ	(240) [1.0953]	LIBRARIANSHIP	(58) [3.0399]
LJUBLJAN	(43) [3.3722]	LJUBLJANA	(46) [3.2989]

Whilst the processing of this query produced 10 relevant Slovene items, only 5 English relevant documents were retrieved. In addition, there was only 1 Slovene retrieved failure reported, compared with 6 failures in the English output. Using the above list of English stems it is fairly easy to explain failures. Both types of stemming errors are present, as follows:

- overstemming: two non-related words PUBLIC (e.g., PUBLIC LIBRARIES) and PUBLICATIONS are conflated to the same stem PUBLIC;
- understemming: LIBRARIANSHIP is not conflated to the stem LIBRARI.

In contrary to Porter's algorithm, the Slovene algorithm was much more effective and did not report any errors in this query. The result is, of course, retrieval of a larger number of Slovene relevant documents.

QUERY 48:

Centralni katalog — avtomatizacija.

Union catalogue — automation.

The processing of this short query produced the following list of stems:

SLOV		ENGL	
CENT	(101) [2.3838]	UNION	(10) [4.9000]
KATAL	(31) [3.7251]	CATALOGU	(30) [3.7600]
AVTOMAT	(24) [3.9957]	AUTOM	(18) [4.2958]

In this case, better performance results were obtained for the English system. While 7 English relevant documents were retrieved, the Slovene system reported retrieval of only 5 relevant items. A less effective performance of the Slovene version of INSTRUCT was caused by overstemming, i.e., a stem CENT– also contained non-related words such as CENTER and CENTRALNI.

The above two examples amply prove that performance of the stemming algorithm can seriously affect multi-lingual output.

Cutoff 20. Experiments concerning the evaluation of IR systems by producing lists of ranked documents have to impose cutoff points. Although the implementation of a ranked cutoff is usually helpful, an artificial distinction within a set of retrieved documents can also occur. In order to find out whether the cutoff factor significantly affects the performance difference between the Slovene and English systems, the ranked cutoff point was extended to 20.

After the top 20 documents from both the Slovene and English lists of retrieved items were analyzed the following numbers of the missing relevant documents were obtained:

- Slovene relevant documents: 32
- English relevant documents: 49

In other words, a total number of relevant documents retrieved by each systems was now as follows (see Table 8.13).

Retrieved documents	Relevant documents
SLOV	334
ENGL	297

Table 8.13: Number of Slovene and English relevant documents at cutoff 20

Table 8.13 demonstrates that—despite the fact that a larger number of the English missing relevant documents were "hidden" within rank 11-20—a performance difference between the Slovene and English systems still remains. To test whether there was a significant difference between both systems, a *sign test* was applied. A value of 0.0031 was obtained; this value was within the region of rejection for $\alpha = 0.05$. On this basis, a null hypothesis stating that there is no difference between the Slovene and English systems, was rejected. In other words, despite the cutoff point being extended to 20, the Slovene version of INSTRUCT still performed significantly better.

However, there is no doubt that these results provide additional insight into the application of statistically-based techniques for multi-lingual IR systems. Having a large percentage of the English relevant documents ranking very close to their Slovene equivalents, the employment of iteration procedures in retrieval (e.g., a relevance feedback search) could significantly improve multi-lingual output. However, in order to test this assumption a much larger document collection—increasing the dispersion of retrieved relevant documents—would be required. A database containing 504 documents as employed in *Experiment II* is too small to provide reliable results about iteration searching in the multi-lingual context.

Recommendations using results of a failure analysis. To summarize, the failure analysis carried out in *Experiment II* has demonstrated that simple statistically-based techniques as implemented within the INSTRUCT package are questionable as a straightforward method in multi-lingual IR systems. The failure analysis found the following two main "barriers":

- natural language (synonyms and other related terms);
- stemming algorithms

However, the very successful performance results obtained by the Slovene version of INSTRUCT indicate that statistically-based techniques could still provide a framework for a multi-lingual IR system. A prerequisite is, of course, that they are enhanced with some other refinements (e.g., a thesaurus look-up process, iteration searching, knowledge-based approach, etc.). It is in this area where additional experimental tests are needed.

Finally—if nothing else—the above results have strongly confirmed that non-conventional, statistically-based techniques, can be introduced in the Slovene operational IR systems. This was additionally proved by the experiment described below.

8.5.2 A multi-lingual experiment based on the identification of word variants

Although the stemming procedure manages to conflate many morphological word variants, INSTRUCT offers further assistance to the end-user who wishes to identify terms that may be useful for the search, and that occur within the document file. This is achieved by the searcher selecting a stem of interest and then allowing the system to identify in the dictionary component of the inverted file those stems which are most similar to the chosen query stem.

The measure of similarity used is based upon the numbers of *trigrams*, i.e., strings of three characters, common to the query stem and each of the stems in the dictionary file. This may be illustrated by the words MIKROFILM and MIKROŽEPEK which give

rise to the stems MIKROFIL– and MIKROŽEP– respectively, and are characterized by the the following two trigram lists ($ denotes the space character):

$MI MIK IKR KRO ROF OFI FIL IL$
$MI MIK IKR KRO ROŽ OŽE ŽEP EP$

The number of trigrams in common (in the above example, there are four identical trigrams) with the query stem is calculated for each of the stems in the database, and these numbers sorted into descending order so as to identify the stems that are most similar to the chosen query. Thus, the submission of the stem MIKROFIL–, from MIKROFILM, results in retrieval and display of the following 10 most similar stems:

1 MIKROFILMAM	(1)	2 MIKROFIS	(1)
3 MIKROPROF	(1)	4 MIKRORAČUNAL	(13)
5 MIKROŽEP	(1)	6 MIKROGRAF	(4)
7 MIKROOB	(4)	8 MIKROPROCESOR	(1)
9 MIKROSNEM	(1)	10 ŽIVIL	(1)

Once the 10 most similar stems have been displayed, the user can select any of them from the list for inclusion in the query.

This means of identifying word variants was first used on a substantial scale by Freund and Willett (1982), and an example of an operational retrieval system that uses the approach is described by Porter (1983). The advantage of the approach is that it permits not just the identification of morphological variants, but also other sorts of variants such as spelling errors, valid alternative spellings, and words with different prefixes.

In *Experiment II*, a test was employed to find out whether this approach can equally well be applied to Slovene and English words. The following two sets of stems were processed:

- 148 Slovene query stems and 2,957 document stems from the Slovene dictionary file;
- 144 English query stems and 3,012 document stems from the English dictionary file.

Displayed index stems (at cutoff 10) were considered to be related to a query stem if they

- had the same basic character structure as the query stem and were semantically related to it, e.g. *COMPUT* and MICROCOMPUT; *SCIENTIF* and SCIENTIST; *DOSTOP* and PRISTOP; *MUZEJ* and MUZEOL; or
- were unmistakable misspellings of the query term which could not have arisen from the misspelling of the another word, e.g. *LIBRARI* and LIBARI; *BIBLIOT* and BIBLIT; *RAČUNAL* and RAČUNALKIK.

(In each case, the query stem has been italized). Similar criteria for the definition of a related term were employed in the experiment carried out by Freund and Willett (1982).

Analysis of results

After all stems from both query sets had been processed by the query expansion module, the following results were obtained, as shown in Table 8.14.

Query set	Stem types	Related stems	Related stems per term
SLOV	148	223	1.5
ENGL	144	240	1.7

Table 8.14: Number of retrieved related stems, using Slovene and English query sets

Table 8.14 shows that the application of the query expansion module, based on a measure of trigram similarity, produced very similar results. Whilst searching in the English dictionary file resulted in 240 related terms (1.7 term per stem), stems from the Slovene query set retrieved 223 related words in the Slovene dictionary file (1.5 per stem).

It is important to note at this point that the large number of non-related terms in both English and Slovene displays is due to the fact that this approach was based on the identification of related stems, and not word variants. Since both stemming algorithms have already managed to reduce morphological variants of words in dictionary files, a trigram similarity approach was trying to identify the following:

- stemming errors (in particular, errors caused by understemming);
- semantically related stems (in particular, stems with different prefixes);
- spelling errors.

According to data in Table 8.15, the identification of stem variants was most effective in the area of retrieving semantically related stems which, in particular, differed in prefixes (e.g., COMMUN retrieved INTERCOMMUN, TELECOMMUN, COMMUNICOLOGI; GRAD retrieved DOGRAD, IZGRAD, NADGRAD, etc.). The set of English stems retrieved 173 (72.1%) semantically related terms; the processing of the Slovene set produced 174 (78.0%) semantically related stems. This demonstrates a great level of similarity between the Slovene and English results.

Identification of:	SLOV	ENGL
1. Stemming errors	35	66
2. Semantically related stems	174	173
3. Spelling errors	14	1

Table 8.15: Identification of related stems

The main difference between the two systems was in the identification of stemming errors. Applying the term expansion module to the stems in the English set of queries resulted in retrieval of 66 terms (27.5%) from the dictionary file which were understemmed. In the Slovene list of retrieved related terms, the percentage of such stems was 15.7% (i.e., 35 stems). The larger percentage of retrieved understemmed terms in

English is mainly due to the characteristics of Porter's algorithm, as emphasized in the previous section (e.g., stems SCIENTIF–, SCIENTIST–, SCIENC– are not conflated to a single root).

In this experiment, it was also interesting to note the performance of the automatic spell-checker. Whilst 14 spelling errors (6.3%) were identified in the Slovene dictionary file, the occurrence of only 1 English spelling error demonstrates the lack of automatic spell-checkers for Slovene.

Despite some of the above performance differences, a query expansion experiment—using a trigram similarity measure—confirmed the initial assumptions. The similar number of semantically related stems that were obtained from dictionary components of the English and Slovene inverted files indicates that this approach is applicable also to the Slovene IR systems. In addition, this type of query expansion could potentially help other statistically-based techniques to become more effective within a multi-lingual IR environment.

8.6 Conclusions

The detailed analysis of the results has confirmed only one of the three main hypotheses. This hypothesis (*HYPOTHESIS 3*) stated that identification of stem variants—using a string similarity measure—will produce a similar number of related terms from both English and Slovene dictionary components of the inverted files. The other two hypotheses (*HYPOTHESIS 1* and *HYPOTHESIS 2*) were rejected, i.e., :

1. Processing of the English documents and queries within a best-match environment did *not* produce more or less identical hits to those retrieved from the Slovene database. The level of only 50% identical hits is sufficient reason to reject *HYPOTHESIS 1*.

2. There was a significant performance difference in retrieving relevant documents. The Slovene version of INSTRUCT produced significantly better results than its English equivalent.

The frequent occurrence of synonyms and other related terms in natural language which are not controlled in INSTRUCT, and automatic word conflation carried out by two different stemming algorithms were detected by a failure analysis as the main reasons for performance differences between the Slovene and English systems. Thus, the results of *Experiment II* provided the following conclusions:

1. The successful performance of the Slovene version of the INSTRUCT package has confirmed that statistically-based techniques can be implemented in the Slovene operational IR systems without any hesitation.

2. Simple statistical techniques—using single term indexing with term weights assignments—are not appropriate as a straightforward method in the multi-lingual IR approach.

However, *Experiment II* provided some indicators that statistically-based IR techniques could still provide a broad framework for multi-lingual retrieval. The prerequisite is that they are enhanced with other refinements. The main problem to be solved in a multi-lingual information system is how to improve the *content* analysis of multi-lingual text. There is no doubt that automatic indexing techniques (assignment of weights to the single index terms) cannot entirely represent the full text content in two languages. Thus, the implementation of a thesaurus look-up process can, for example, potentially improve multi-lingual retrieval effectiveness.

Some other retrieval techniques can also be considered as promising ways to improve a multi-lingual output. Whilst some of them are also based on statistical principles (for example, iteration searching – relevance feedback search, query expansion modules), the others derive from the knowledge-based environment. However, the effect of these techniques on the improvement of a simple statistically-based approach in a multi-lingual IR system has not been tested within *Experiment II*. Thus, any recommendations for combining these techniques with single term indexing—in order to improve a multi-lingual output—remain speculative.

Chapter 9

Conclusions

9.1 Introduction

The primary aim of this project, which started in the academic year 1988/89, was to facilitate end-user access to bibliographic databases in Slovenia. At present, the information retrieval environment in Slovenia is characterized by the following:

- the growing number of bibliographic and other types of databases;
- increasing user demands for accurate and up-to-date information within a multilingual context;
- the application of different information retrieval systems.

It is important to note that all software systems (e.g, ATLASS, TRIP) available for accessing these databases are typical of current retrieval software elsewhere in that they are based on Boolean searching, with professional intermediaries being used to carry out on-line searches on behalf of end-users. Furthermore, the effectiveness and efficiency of these systems have rarely been evaluated. Consequently, modern, non-conventional methods and techniques of information retrieval which allow direct end-users interaction with the system are neither incorporated into existing retrieval systems in Slovenia, nor has much research been carried out in this area.

This is, of course, a very questionable situation because the provision of end-user searching facilities has been recognized in many countries as the only way to remove a barrier between the original source of a query and the query's answer.

One of the main research areas in information retrieval whose main aim is to enable end-users to carry out searching in both an efficient and effective manner is based on the development of algorithmic procedures which allow the computer to undertake many of the functions of a trained intermediary. This approach, based on the use of a range of statistical techniques, is also known as *statistically-based* retrieval.

Many such document retrieval systems have been described in the research literature and operational implementations of some of these ideas are now available (Willett, 1988a). To date, the great bulk of this work has been carried out with English language material, where the necessary linguistic facilities, i.e., stop-word lists and stemming routines, have been available for many years (Lovins, 1968). Therefore, the main problem which was investigated in the context of this PhD project, is contained in the following two questions:

1. Are statistically-based techniques applicable to Slovene information retrieval systems?

2. Could statistically-based techniques provide a framework for developing multilingual information retrieval systems?

The second point is of particular importance to end-users in Slovenia, who are surrounded by document collections written in different languages (i.e., Yugoslav languages and other major European languages).

9.2 Summary of results and conclusions

9.2.1 Development of a stop-word list and a stemming algorithm

The use of best-match searching techniques in a Slovene information retrieval environment was tested by the employment of the INSTRUCT (INteractive System for

Teaching Retrieval Using Computational Techniques). The processing routines in INSTRUCT are, in very large part, independent of the actual language in which the text have been written. However, the implementation of a Slovene language-based information retrieval system required development of the following two language-dependent components:

- the creation of a general purpose stop-word list;
- the design of a powerful stemming algorithm which takes account of the language's morphological structure.

The main feature of the Slovene language in the context of a system for automatic word conflation is that new word forms are created by adding derivational and inflectional suffixes to a basic stem. Thus, as with English, many distinctive words with similar meanings can be created from a single stem and it should be possible to implement a conflation procedure for these morphological variants by procedures that utilize a set of suffixes. However, a detailed analysis of the morphological structure of the Slovene language revealed the following:

- the Slovene language exhibits an extremely rich inflectional morphology in both verbal and nominal systems; for example, the word root RAZISKOVA (RESEARCH) can occur in any one of no less than 94 different forms;
- in addition, Slovene is characterized by various types of morphemic alternations, occurring in both stems and suffixes during the inflection.

This implies that an effective stemming algorithm for Slovene text is likely to require many more suffixes and many more complex context-sensitive and recoding rules than is the case with English. In addition, it is extremely difficult to establish iteration patterns; thus, the use of a longest-match algorithm was studied. However, the main aim of the design process was to obtain a reasonable balance between, on the one hand, the number of rules, and on the other hand, simplicity and efficiency of processing.

The starting point for the design of both a stop-word list and a stemming algorithm was data about the general frequency characteristics of Slovene as extracted by extensive study of two Slovene text corpora. The frequency of occurrence of the word types in these text corpora followed a typical Zipfian distribution, with a very few word types providing a very high percentage of the observed tokens. Although these characteristics were analogous to those of English (and many other languages) it was not possible to identify words for inclusion in a stop-word list merely by taking account of the most frequently occurring words (as is commonly done in the case of stop-word list for English databases). Such a procedure would ignore a very important difference between Slovene and English, this being the much greater number of distinct word types that are encountered in the former's natural language text.

Many of the large numbers of low-frequency Slovene words are morphological variants of very commonly occurring function words that certainly should be included in a stop-word list; equally certainly, these low frequency word variants will not be included merely by selecting the most frequently occurring words. The production of a stop-word list for the Slovene language thus entailed a much greater level of detailed, manual involvement than is required for the construction of a stop-word list for the English language.

The resulting stop-word list contained a total of 1,593 non-content bearing words, these consisting of function words such as prepositions, pronouns, auxiliary verbs, conjunctions, etc., together with a small core of other types of terms carrying extremely low meaning in phrases or sentences. It should be noted that this stop-word list can be described as the *first* general purpose stop-word list created for the Slovene language. The results of the evaluation—a comparable level of compression as obtained when similar procedures were applied to English texts, and a successful level of indexing—demonstrated the potential applicability of this list to any information retrieval system in Slovenia.

The first step towards the design of an effective *automatic conflation* procedure for Slovene text was the development of a simple context-free, stemming algorithm,

in which the most frequently occurring endings from the sorted list of reversed words were chosen as the suffixes. No recoding or context-sensitive rules were used, the only constraint on suffix removal being that the remaining stem should not contain less than three characters. This approach was clearly crude in concept but avoided the need for detailed manual processing that characterizes most other ways of creating lists of suffixes. However, the performance of this algorithm was far from satisfactory. The best overall results were obtained with the list containing 2,000 suffixes; even here, however, less than 40% of the words were conflated to the correct root. The poor level of performance meant that a more complex, context-sensitive algorithm needed to be developed.

This algorithm was developed using the traditional, trial-and-error approach that characterizes most context-dependent algorithms. Consideration was given as to the minimum stem length which should be left after the removal of a given ending, of new endings which needed to be added to the suffix list or endings which needed to be removed from it, and of the context-sensitive and recoding rules needed for accurate conflation. It was often the case that selection of one suffix would require the adoption or removal of other suffixes, or the addition of context-sensitive rules in order to maintain consistency; this behavior is, of course, characteristic of all languages and not specific to Slovene.

The resulting longest-match, context-sensitive algorithm is based on the use of 5,276 endings, each of which has an associated minimum stem length, either three or four characters, and one of eight action codes, which implement the context-sensitive rules. In addition, there are three types of recoding rule that are applied after suffix deletion.

The effectiveness of this algorithm was tested in two phases. First, the stemming algorithm was applied to a large text corpus which contained 2,616 distinct word types. In this context, the level of compression and the success rate of suffix stripping were measured. In the second phase, the stemming algorithm was implemented within a best-match information retrieval system, and its retrieval performance evaluated.

If the level of compression is expressed in terms of the number of reduced words, then 54.7% compression was achieved by the employment of this algorithm, demonstrating that it is a *strong* stemmer. However, this did not seem to adversely affect performance since a detailed inspection of resulting stems—as carried out by a trained intermediary—revealed that the success rate of suffix stripping was 90.8%.

Results of this simple test have indicated that the procedures used in the stemming algorithm are workable and will yield good results with only minor changes. Although these alterations might involve the list of endings and occasionally the context-sensitive and recoding rules, the basic principles of the new stemming algorithm remain the same. However, in order to obtain the final results about its *retrieval performance*, the second phase of the experiment was carried out.

9.2.2 Retrieval effectiveness of the stemming algorithm

The retrieval performance of the stemming algorithm was tested on the basis of its employment in a Slovene language version of the text retrieval system INSTRUCT. The Slovene version of INSTRUCT was designed by the conversion of the original (PRIME) version of INSTRUCT to an IBM PC-compatible microcomputer using the TURBO PASCAL 5.5 programming language. This means that the Slovene version of INSTRUCT consists of the following main modules:

- natural language query input (in Slovene);
- elimination of non-content bearing terms from the query (using the dictionary of 1,593 Slovene stop-words);
- stemming of remaining query terms (using the Slovene stemming algorithm);
- morphological term expansion using a string similarity measure;
- best-match searching (with the possibility of imposing Boolean constraints after the initial search has been carried out);
- relevance feedback searching;

- Boolean searching.

The major alterations to the original version of INSTRUCT were carried out in the language-dependent modules in order to achieve the main goal, i.e., the successful processing of Slovene terms both in queries and in documents.

The performance effectiveness of the stemming algorithm was tested by its comparison with two other types of text representation: manual right-hand truncation, carried out by a trained intermediary, and non-stemming. Searches were carried out using these three types of text representation on a specially-created Slovene document test collection—the first such collection to be created—which contained 504 documents in the library and information science field, a set of 48 queries, and relevance judgments of retrieved documents.

The retrieval effectiveness of the three different types of search was tested by applying well-known measures of recall and precision. In addition, the statistical significance of the differences were tested using the sign test and the Kendall coefficient of concordance, W.

The results of the comparative evaluation of the three different types of search revealed the following:

- there is a significant performance difference between automatic word conflation and unstemmed processing of the Slovene text;
- there is no significant performance difference between automatic stemming and manual right-hand truncation, carried out by a trained intermediary.

It follows that one of the important components of an information retrieval system, i.e., word conflation, can be automated in Slovene systems with no average loss of performance, thus allowing users easier access to the systems.

It is also interesting to note that the quite huge difference between stemmed and unstemmed searches was caused mainly by the richness of the morphology of Slovene. Similar searches on English databases (see, for example, Harman, 1991) suggest that

automatic word conflation achieves only a slight improvement. This is an additional argument for the importance of an effective stemming algorithm within a Slovene information retrieval environment.

9.2.3 Multi-lingual approach to document retrieval

Having obtained good performance results with the employment of the Slovene stemming algorithm, the next experiment (*Experiment II*) was carried out. Its main objective was to test the performance of statistically-based techniques in two different languages, i.e., Slovene and English. It was hoped that the results of this experiment could serve as an important contribution towards solving the problem of a multi-lingual approach to document retrieval in Slovenia.

Experiment II sought to accomplish the following two main tasks; firstly, to test whether a Slovene information retrieval system—using statistically-based techniques—can achieve the analogous retrieval effectiveness as obtained by an English system; and, secondly, to examine whether statistically-based techniques are suitable in designing multi-lingual information systems. The same methodology as in the previous experiment was employed, i.e., a laboratory test was carried out, using the Slovene test collection and its English equivalent. Apart from measuring the retrieval performance of both systems, a detailed failure analysis was also employed. In addition, a Slovene and an English system were compared by their capability of producing semantically related stems to the selected query stem. As with *Experiment I*, this was the first such experiment carried out in Slovenia, and was also one of very few similar projects reported, so far, worldwide.

A detailed analysis of performance results—expressed in terms of number of identical hits and by measuring recall and precision—has confirmed only one of three main hypotheses. Although the experiment on the identification of stem variants produced a similar number of related terms from both English and Slovene dictionary components of the inverted file, the other two hypotheses were rejected, i.e.:

- processing of the English documents and queries did not produce more or less

identical hits to those retrieved from the Slovene database;

- the Slovene version of INSTRUCT produced significantly better performance results than its English equivalent.

These results were unexpected, and were investigated by the failure analysis. This showed that the differences were due to the frequent occurrence of synonyms and other related terms, and to the behaviour of the two different stemming algorithms. This raises questions as to the suitability of statistically-based techniques for developing multi-lingual information systems, as implemented here.

However, better results might be obtained by enhancing the simple strategies tested here with some other refinements such as a thesaurus look-up process, iteration searching, a knowledge-based approach, etc. However, the effect of these techniques on the improvement of a simple statistically-based approach in a multi-lingual retrieval environment has not been tested in *Experiment II*, mainly because of the small size of the test collection available to us.

9.3 Suggestions for further work

There is no doubt that this PhD project has clearly demonstrated the applicability of statistically-based retrieval techniques to a Slovene information retrieval environment. Apart from developing some other language-dependent techniques (e.g, a Slovene equivalent to the Soundex code), the following are some suggestions which could lead to the increased use of advanced information retrieval techniques in Slovenia:

- INSTRUCT as a teaching resource in information retrieval courses at the Department of Librarianship, University of Ljubljana;
- INSTRUCT as a demonstration package installed at the Computing Centre, University of Maribor, which acts as a host for a large number of databases;
- incorporation of some INSTRUCT modules into existing Slovene retrieval packages, particularly those still under development.

However, it should be noted that these advanced techniques of information retrieval will be firmly established in Slovenia only if they can be enhanced with refinements which allow a multi-lingual approach to document retrieval. There are only two million people living in Slovenia, and end-users are faced with document collections written not only in Slovene, but also either in other Yugoslav languages or in other major European languages. One of the main requirements of end-users in Slovenia will be to be able to input a query and to receive a query answer in Slovene language from any of these databases. It is clear from the results of this thesis that the provision of such facilities will require a very large amount of research.

Appendix A

A list of consulted literature

Bidwell, C.A. (1969) *Outline of Slovenian Morphology.* Pittsburgh: University of Pittsburgh.

Hamp, E.P. (1975) On the dual inflections in Slovene. *Slavistična revija*, Vol. 23, pp. 67-70.

Lencek, R.L. (1966) *The Verb Pattern of the Contemporary Standard Slovene.* Wiesbaden: Otto Harrassowitz.

Lencek, R.L. (1982) *The Structure and History of the Slovene Language.* Columbia: Slavica.

Paternost, J. (1963) *The Slovenian Verbal System: Morphophonemics and Variations.* PhD Thesis, Indiana University.

Rigler, J. (1966) Premene tonemov v oblikoslovnih vzorcih slovenskega knjižnega jezika. *Jezik in slovstvo*, Vol. 10, pp. 24-35.

Tollefson, J.W. (1981) *The Language Situation and Language Policy in Slovenia.* Washington: University Press of Slovenia.

Toporišič, J. (1966) Esej o slovenskih besednih vrstah. *Jezik in slovstvo*, Vol. 10, pp. 295-305.

Toporišič, J. (1967) Strukturiranost slovenskih glasov in predvidljivost njihove razvrstitve. *Jezik in slovstvo*, Vol. 11, pp. 92-96.

Toporišič, J. (1975) Main characteristics of the Slovene language. *In*: Komac, D. and Škerlj, R. *English-Slovene and Slovene-English Dictionary.* Ljubljana: Cankarjeva založba, pp. 417-435.

Toporišič, J. (1978) A language of a small nationality in a multilingual state. *Folia Slavica*, Vol.1, pp. 480-487.

Toporišič, J. (1984) *Slovenska slovnica.* 2nd ed., Maribor: Obzorja.

Vidovič-Muha, A. (1988) *Slovensko skladenjsko besedotvorje ob primerih zloženk.* Ljubljana: Partizanska knjiga.

Appendix B

The list of natural language queries

The list of queries consists of 48 queries as provided by 8 users.

1. Marketing (trženje) v knjižnicah, identiteta knjižnice in stiki z javnostjo.
2. Knjižnice in informacijski centri v Veliki Britaniji in Indiji.
3. Selektivna diseminacija informacij (SDI) in retrospektivne poizvedbe (RP).
4. Klasificiranje (UDK - univerzalna decimalna klasifikacija), klasifikacijske sheme in klasifikacijski sistemi.
5. Znanstveno-raziskovalno delo in razstavna dejavnost Referalnega centra Univerze v Zagrebu.
6. Specializirani INDOK centri v Sloveniji in na Hrvaškem.
7. Informacijsko-dokumentacijski centri (INDOK) v Jugoslaviji.
8. Specializirani INDOK centri (SIC) v Sloveniji in Jugoslaviji.
9. Računalniški programi (programska oprema, programski paketi in sistemi, softver) na področju knjižničarstva in dokumentalistike.
10. Indeksiranje (dokumentiranje) in sekundarni dokumenti (sekundarne publikacije) v knjižnično-informacijskem sistemu (KIS) in sistemu znanstvenih in tehničnih informacij (SZTI).
11. Mednarodni standardni bibliografski opis (ISBD) in UNISIST v Sloveniji in Jugoslaviji.
12. Splošnoizobraževalno knjižničarstvo v Ljubljani.

13. Zaščita in izboljšanje človekovega okolja - pomen informacij in dokumentov ter delovanje INDOK centra v Zagrebu.

14. Pomen bibliometrije in analize citatov pri evalvaciji kvalitete znanstvenih del in časopisov.

15. Muzeologija, muzejska dejavnost in muzejski eksponati.

16. Raziskovalna dejavnost na področju bibliotekarstva in informatike v Sloveniji ter razvoj bibliotekarske vede (stroke).

17. Modeli za določanje zanesljivosti softvera.

18. Teorija in modeli samoupravnega javnega komuniciranja.

19. Vsebinska (predmetna) obdelava in avtomatizacija.

20. Problematika iskanja informacij (poizvedbe).

21. Standardi v knjižničarstvu in knjižnicah.

22. Mikroračunalniki v knjižnicah in INDOK centrih.

23. Obvezni izvod in slovenska nacionalna bibliografija.

24. Domoznanske zbirke.

25. Izobraževanje (šolanje, vzgoja) knjižničarskih kadrov (bibliotekarjev, bibliotekarskih kadrov) - učni načrti in strokovni izpiti.

26. Projektiranje (planiranje, načrtovanje) bibliotečnih (knjižničnih) stavb - gradnja (izgradnja), prostori, notranja oprema in ureditev.

27. Zakonodaja in zakoni o knjižničarstvu v Sloveniji.

28. Delo z bralci, knjižna vzgoja, branje v šolskih, pionirskih in mladinskih knjižnicah.

29. Povezovanje (sodelovanje) šolskih (pionirskih, mladinskih) knjižnic s splošnoizobraževalnimi knjižnicami (SIK-i).

30. Časniki (časopisi, serijske oz. periodične publikacije) in mikrofilmanje.

31. Informacijska služba v knjižnicah.

32. Podatkovne zbirke (baze podatkov) v družboslovnih in humanističnih vedah.

33. Medbibliotečna (medknjižnična) izposoja.

34. Bralci in uporabniki v splošnoizobraževalnih knjižnicah.

35. Citatna analiza (analiza citatov).

36. Karkoli o Narodni in univerzitetni knjižnici v Ljubljani.

37. Računalniška obdelava (avtomatizacija) nacionalnih bibliografij.

38. Metode poizvedovanja (iskanja) v bibliografskih bazah podatkov (podatkovnih zbirkah).

39. Analiza uporabnikov v knjižnicah.

40. Pomen mikrofilma in mikrofilmanja v knjižnicah in INDOK centrih.

41. Specializirane baze podatkov (podatkovne zbirke) v Jugoslaviji.

42. Avtomatizacija poslovanja nacionalnih knjižnic.

43. Knjižnice - koordinacija - nabavna politika (nakupi).

44. Publikacije in informacije - univerzalna (splošna) dostopnost.

45. Bibliografije - katalogi - normativna kontrola - normativne datoteke.

46. Univerzitetne (univerzne, visokošolske) knjižnice - standardi.

47. Predmetni katalog - indeksiranje - tezaver.

48. Centralni katalog - avtomatizacija.

Appendix C

The list of queries as processed by the trained intermediary

The order of this list corresponds to the list in Appendix B. The question mark '?' indicates at which point the trained intermediary removed an ending from the word.

1. Market? (trg? trž?) knjižni? identi? knjižni? stik? javnost?
2. Knjižni? informac? cent? Velik? Britan? Indij?
3. Selektiv? disemin? informacij? (SDI) retrospekt? poizved? (RP)
4. Klasifi? (UDK, univerzaln? decimaln? klasifi?) klasifi? shem? klasifi? sistem?
5. Znanstven? raziskoval? del? razstav? dejavn? Referaln? cent? Univerz? Zagreb?
6. Special? INDOK cent? Sloven? Hrvašk? (Hrvat?)
7. Informac? dokument? cent? (INDOK) Jugosl?
8. Special? INDOK cent? (SIC) Sloven? Jugosl?
9. Računaln? program? (program? oprem? program? paket? sistem? softver?) knjižni? dokument?
10. Indeks? (dokument?) sekund? dokument? (publik?) knjižn? informac? sistem? (KIS) sistem? znanstv? tehn? informacij? (SZTI)
11. Mednarodn? standard? bibliografsk? opis? (ISBD) UNISIST Slovenij? Jugosl?
12. Splošnoizobraževaln? knjižničar? Ljubljan?
13. Zaščit? izboljš? človek? okol? pomen inform? dokument? delov? INDOK cent? Zagreb?

14. Pomen bibliomet? anali? citat? evalv? kvalit? znanstv? del časopis?

15. Muzeol? muzej? dejavnost? muzej? eksponat?

16. Raziskov? dejavnost? področ? bibliotekar? informatik? Slovenij? razvoj bibliotekars? ved? (strok?)

17. Model? določ? zanesljivost? softver?

18. Teor? model? samoupravn? javn? komunic?

19. Vsebin? (predmet?) obdel? avtomat?

20. Problem? iskan? inform? (poizved?)

21. Standard? knjižni?

22. Mikroračun? knjižni? INDOK cent?

23. Obvezn? izvod? slovensk? nacional? bibliograf?

24. Domoznan? zbirk?

25. Izobra? (šolan? vzgoj?) knjižni? kad? (bibliotekar? kad?) uč? načrt? strokov? izpit?

26. Projekt? (plan? načrt?) bibliot? (knjižni?) stavb? grad? (izgrad?) prosto? notran? oprem? uredit?

27. Zakon? knjižni? Sloven?

28. Delo? bral? knjižn? vzgoj? bran? šolsk? pionir? mladin? knjižni?

29. Povez? (sodel?) šolsk? (pionir? mladin?) knjižni? splošnoizobraževal? knjižni? (SIK)

30. Časnik? (časopis? serij? period? publik?) mikrofilm?

31. Inform? služb? knjižni?

32. Podatk? zbir? (baz? podatk?) družboslov? humani? ved? (znan?)

33. Medbibliotečn? (medknjižničn?) izposoj?

34. Bral? uporabn? splošnoizobraževaln? knjižni?

35. Citat? anali?

36. Narodn? univerz? knjižnic? Ljubljan? (NUK)

37. Računaln? obdel? (avtomat?) nacion? (narod?) bibliograf?

38. Metod? poizved? (iskan?) bibliograf? baz? (podatk? zbirk?)

39. Anali? uporabn? knjižn?

40. Pomen mikrofilm? knjižni? INDOK centr?

41. Special? baz? podatk? (podatk? zbir?) Jugosl?

42. Avtomat? poslov? nacionaln? knjižni?

43. Knjižni? - koordin? nabav? politik? (nakup?)

44. Publi? inform? - univerzaln? (splošn?) dostop?

45. Bibliograf? - katalog? - normat? kontrol? - normat? datotek?

46. Univerzitet? (univerz? visokošol?) knjižni? - standard?

47. Predmet? katalog? - indeks? -tezav?

48. Centraln? katalog? - avtomat?

Appendix D

The list of English language queries

This list corresponds to the list of queries in Slovene language, as presented in Appendix B.

1. Marketing in libraries, library's identity and public relation.
2. Libraries and information centres (centers) in Great Britain and India.
3. Selective dissemination of information (SDI) and retrospective searches.
4. Classification (UDC - Universal Decimal Classification), classification schemes, classification systems.
5. Research work and exhibition activities of the Referral Centre of the University of Zagreb.
6. Specialized INDOC centres (centers; information and documentation services) in Slovenia and Croatia.
7. Information and documentation services (INDOC centres; centers) in Yugoslavia.
8. Specialized INDOC centres (centers; information and documentation services) in Slovenia and Yugoslavia.
9. Software (software packages and systems, program packages) in librarianship and documentation.
10. Indexing (documentation) and secondary documents (secondary publications) in library and information services and in scientific and technical information services.
11. International Standard Bibliographic Description (ISBD) and UNISIST in Slovenia and Yugoslavia.

12. Public librarianship in Ljubljana.
13. Protection and improvement of the human environment — the importance of information and documents and activity of the INDOC Centre in Zagreb.
14. Bibliometrics and citation analysis in evaluation of scientific papers and periodicals.
15. Museology, museum activities, and museum exhibits.
16. Research work in the field of librarianship and information science in Slovenia and development of library science (profession).
17. Software reliability models.
18. Theory and models of self-management mass communication.
19. Subject (content) analysis and automation.
20. Information retrieval and searching.
21. Standards in libraries and librarianship.
22. Microcomputers in libraries and INDOC centres (centers; information and documentation services).
23. The deposit copy and the Slovene National Bibliography.
24. Local (ethnographic, demographic) collections.
25. Training and education of librarians (library staff, personnel) — educational programmes (curriculums) and examination regulations.
26. Planning and design of library buildings — construction, space requirements, interior equipment and layout.
27. Legislation and legal acts to do with librarianship and libraries in Slovenia.
28. Book and literary education, readers and reading in school, pioneers' and youth libraries.
29. Cooperation (co-operation) between school (pioneers', youth) libraries and public libraries.
30. Newspapers (periodicals, journals, serial publications) and microfilming.
31. Information services in libraries.
32. Data bases (databases) in social sciences and humanities.
33. Interlibrary lending (inter-library loan).
34. Readers and users in public libraries.

35. Citation analysis.

36. Anything about the National and University Library in Ljubljana.

37. Computer-based (automated) processing of national bibliographies.

38. Information retrieval methods for searching in bibliographic databases (data bases, collections).

39. User studies (surveys) in libraries.

40. Microfilm and microfilming in libraries and INDOC centres (centers; information and documentation services).

41. Specialized databases (data bases, collections) in Yugoslavia.

42. National libraries and automation.

43. Libraries — co-ordination (coordination) — acquisition policy.

44. Publications and information — universal (general) availability.

45. Bibliography — catalogues — authority control — authority files.

46. University libraries — standards.

47. Subject catalogue — indexing — thesaurus.

48. Union catalogue — automation.

REFERENCES

Al-Hawamdeh, S. and Willett, P. (1989) Paragraph-based nearest neighbour searching in full-text documents. *Electronic Publishing*, Vol. 2, pp. 179-192.

Angell, R.C. *et al.* (1983) Automatic spelling correction using a trigram similarity measure. *Information Processing and Management*, Vol. 19, pp. 255-261.

Ashford, J. and Willett, P. (1988) *Text Retrieval and Document Databases*. Bromley: Chartwell-Bratt.

Bar Hillel, Y. (1962) Theoretical aspects of the mechanization of literature searching. *In*: W. Hoffman (*ed*). *Digitale Informationswandler*. Braunschweig: Vieweg and Sons, pp. 406-443.

Bawden, D. (1986) Information systems and the stimulation of creativity. *Journal of Information Science*, Vol. 12, pp. 203-216.

Bawden, D. (1990) *User-oriented Evaluation of Information Systems and Services*. London: Gower.

Bell, C.L. and Jones, K.P. (1976) A minicomputer retrieval system with automatic root finding and roling facilities. *Program*, Vol. 10, pp. 14-27.

Bidwell, C.A. (1969) *Outline of Slovenian Morphology*. Pittsburgh: University of Pittsburgh.

Biru, T. *et al.* (1989) Inclusion of relevance information in the term discrimination model. *Journal of Documentation*, Vol 45, pp. 85-109.

Booth, A.D. (1967) A "law" of occurrences for words of low frequency. *Information and Control*, Vol. 10, pp. 386-393.

Brzozowski, J.P. (1983) MASQUERADE: searching the full text of abstracts using automatic indexing. *Journal of Information Science*, Vol. 6, pp. 67-73.

Carroll, D.M. *et al.* (1988) Bibliographic pattern matching using the ICL Distributed Array Processor. *Journal of the American Society for Information Science*, Vol. 39, pp. 390-399.

Cercone, N. (1978) Morphological analysis and lexicon design for natural-language processing. *Computers and the Humanities*, Vol. 11, pp. 235-258.

Chiaramella, Y. and Defude, B. (1987) A prototype of an intelligent system for information retrieval: IOTA. *Information Processing and Management*, Vol. 23, pp. 285-303.

Cleverdon, C.W. (1966) *Factors Determining the Performance of Indexing Systems*. Cranfield: College of Aeronautics.

Cleverdon, C. (1984) Optimizing convenient online access to bibliographic databases. *Information Services and Use*, Vol. 4, pp. 37-47.

Cooper, D. and Lynch, M.F. (1979) Compression of Wiswesser line notations using variety generation. *Journal of Chemical Information and Computer Sciences*, Vol. 19, pp. 165-169.

Croft, W.B. and Harper, D.J. (1979) Using probabilistic models of document retrieval without relevance information. *Journal of Documentation*, Vol. 35, pp. 285-295.

Cuadra, C.A. and Katter, R.V. (1967) Opening the black box of "relevance". *Journal of Documentation*, Vol. 23, pp. 291-303.

Dawson, J.L. (1974) Suffix removal and word conflation. *ALLC Bulletin*, Vol. 2, pp. 33-46.

Dimec, J. (1988) *Računalniška analiza slovenskega jezika v medicini (A Computer Analysis of Slovene Language in Medicine).* M.Sc. thesis, University of Ljubljana.

Dolby, J.L. and Resnikoff, H.L. (1964) On the structure of written English. *Language*, Vol. 40, pp. 167-196.

Doszkocs, T.E. (1983) CITE NLM: natural language searching in an online catalog. *Information Technology and Libraries*, Vol. 2, pp. 364-380.

Dubois, C.P.R. (1979) Multilingual information systems: Some criteria for the choice of specific techniques. *Journal of Information Science*, Vol. 1, pp. 5-12.

Ellis, D. (1987) *The Derivation of a Behavioural Model for Information Retrieval System Design.* Ph.D thesis, University of Sheffield.

Ellis, D. (1990) *New Horizons in Information Retrieval.* London: The Library Association.

Fagan, J.L. (1989) The effectiveness of a nonsyntactic approach to automatic phrase indexing for document retrieval. *Journal of the American Society for Information Science*, Vol. 40, pp. 115-132.

Field, B.J. (1975) *Semi-automatic Development of Thesauri Using Free-language Vocabulary Analysis (Part 1 only).* Report no. R75/24 : INSPEC.

Field, B.J. (1977) Automatic indexing for multilingual systems. *Third European Congress on Information Systems and Networks: Overcoming the Language Barrier.* London: Saur, pp. 469-492.

Frakes, W.B. (1984) Term conflation for information retrieval. *In:* C.J. van Rijsbergen (*ed*). *Research and Development in Information Retrieval.* Cambridge: CUP, pp. 383-390.

Freund, G.E. and Willett, P. (1982) Online identification of word variants and arbitrary truncation searching using a string similarity measure. *Information Technology: Research and Development*, Vol. 1, pp. 177-187.

Fuhr, N. (1990) Zur Überwindung der Diskrepanz zwischen Retrievalforschung und -praxis. *Nachrichten für Dokumentation*, Vol. 41, pp. 3-7.

Goldsmith, N. (1982) An appraisal of factors affecting the performance of text retrieval systems. *Information Technology: Research and Development*, Vol. 1, pp. 41-53.

Griffiths, A. *et al.* (1984) Hierarchic agglomerative clustering methods for automatic document classification. *Journal of Documentation*, Vol. 40, pp. 175-205.

Griffiths, A. *et al.* (1986) Using interdocument similarity information in document retrieval systems. *Journal of the American Society for Information Science*, Vol. 37, pp. 3-11.

Hafer, M.A. and Weiss, S.F. (1974) Word segmentation by letter successor varieties. *Information Storage and Retrieval*, Vol. 10, pp. 371-385.

Harman, D. (1987) A failure analysis on the limitations of suffixing in an online environment. *Proceedings of the Tenth International Conference on Research and Development in Information Retrieval.* Washington: ACM, pp. 102-108.

Harman, D. (1991) How effective is suffixing? *Journal of the American Society for Information Science*, Vol. 42, pp. 7-15.

Harter, S.P. (1975) A probabilistic approach to automatic keyword indexing. Part II. An algorithm for probabilistic indexing. *Journal of the American Society for Information Science*, Vol. 26, pp. 280-289.

Hendry, I.G. *et al.* (1986a) INSTRUCT: a teaching package for experimental methods in information retrieval. Part 1. The users' view. *Program*, Vol. 20, pp. 245-263.

Hendry, I.G. *et al.* (1986b) INSTRUCT: a teaching package for experimental methods in information retrieval. Part 2. Computational aspects. *Program*, Vol. 20, pp. 129-151.

Hildreth, C.R. (1982) Online browsing support capabilities. *Proceedings of the ASIS Annual Meeting 19*. White Plains, New York: Knowledge Industry Publications Inc., pp. 127-132.

Institute for Information Science, University of Maribor (1990) *ATLASS in sistem vzajemne katalogizacije*. Maribor: University of Maribor.

Jäppinen, H. *et al.* (1985) FINNTEXT—text retrieval system for an agglutinative language. *RIAO 85 Recherche d'Informations*, Grenoble, pp. 217-226.

Jones, K.P. and Bell, C.L.M. (1984) The automatic extraction of words from text especially for input into information retrieval systems based on inverted files. *In*: C.J. van Rijsbergen (*ed.*) *Research and Development in Information Retrieval*. Cambridge: CUP, pp. 409-419.

Keen, E.M. (1991a) The use of term position devices in ranked output experiments. *Journal of Documentation*, Vol. 47, pp. 1-22.

Keen, E.M. (1991b) The effect of stemming strength on the effectiveness of output ranking. Paper given at Informatics 11, March 1991, 13 p.

Kimberley, R. (*ed.*) (1987) *Text Retrieval: A Directory of Software. 2nd edition.* Aldershot: Gower.

Kosmač, C. (1953) *Pomladni dan.* Ljubljana: Državna založba.

Lancaster, F.W. (1969) MEDLARS: Report on the evaluation of its operating efficiency. *American Documentation*, Vol. 20, pp. 119-142.

Lencek, R.L. (1966) *The Verb Pattern of the Contemporary Standard Slovene.* Wiesbaden: Otto Harrassowitz.

Lencek, R.L. (1982) *The Structure and History of the Slovene Language.* Columbia: Slavica.

Lennon, M. *et al.* (1981) An evaluation of some conflation algorithms for information retrieval. *Journal of Information Science*, Vol. 3, pp. 177-183.

Lesk, M.E. and Salton, G. (1969) Relevance assessments and retrieval system evaluation. *Information Storage and Retrieval*, Vol. 4, pp. 343-359.

Lovins, J.B. (1968) Development of a stemming algorithm. *Mechanical Translation and Computational Linguistics*, Vol. 11, pp. 22-31.

Lovins, J.B. (1971) Error evaluation for stemming algorithms as clustering algorithms. *Journal of the American Society for Information Science*, Vol. 22, pp. 28-40.

Lowe, T.C. *et al.* (1973) *Additional Text Processing for On-line Retrieval (The RADCOL System).* Technical Report RADC-TR-73-337.

Luhn, H.P. (1957) A statistical approach to mechanised encoding and searching of library information. *IBM Journal of Research and Development*, Vol.1, pp. 309-317.

Luhn, H.P. (1958) The automatic creation of literature abstracts. *IBM Journal of Research and Development*, Vol. 2, pp. 159-165.

Lynch, M.F. (1977) Variety generation - A reinterpretation of Shannon's mathematical theory of communication and its implications for Information Science. *Journal of the American Society for Information Science*, Vol. 28, pp. 19-24.

Marcus, R.S. (1983) An experimental comparison of the effectiveness of computers and humans as search intermediaries. *Journal of the American Society for Information Science*, Vol. 34, pp. 381-404.

Markey, K. (1983) *Online Catalogue Use: Results of Surveys and Focus Group Interviews in Several Libraries. Vol. II.* OCLC Online Computer Library Center.

Martinović, S. (1985) Automatizovani višejezični tezaurus. *Informatika*, Vol. 13, pp. 21-35.

McCain K.W. *et al.* (1987) Comparing retrieval performance in online data bases. *Information Processing and Management*, Vol. 23, pp. 539-553.

McCall, F.M. and Willett, P. (1986) Criteria for the selection of search strategies in best match document retrieval systems. *International Journal of Man-Machine Studies*, Vol. 25, pp. 317-326.

Mohan, K.C. (1987) *Choice of Retrieval Techniques for a Multi-Strategy Retrieval System.* PhD thesis, Sheffield: University of Sheffield.

Murtagh, F. (1983) A survey of recent advances in hierarchical clustering algorithms.

The Computer Journal, Vol. 26, pp. 354-359.

Nelis, K. (1985) *Human Interaction with Computers in an Information Retrieval Context: a study of the Users' Interaction with INSTRUCT*. MSc Dissertation, Sheffield: University of Sheffield.

Niedermair, G.T. *et al.* (1985) MARS: a retrieval tool on the basis of morphological analysis. *In*: C.J. van Rijsbergen (*ed*). *Research and Development in Information Retrieval.* Cambridge: CUP, pp. 369-380.

Noreault, T. and Chatham, R. (1982) A procedure for the estimation of term similarity coefficients. *Information Technology: Research and Development*, Vol. 1, pp. 189-196.

Noreault, T. *et al.* (1977) Automatic ranked output from Boolean searches in SIRE. *Journal of the American Society for Information Science*, Vol. 28, pp. 333-339.

Overhage, C.F.J. and Reintjes, J.F. (1974) Project INTREX: A general review. *Information Storage and Retrieval*, Vol. 10, pp. 157-188.

Pape, D.L. and Jones, R.L. (1988) STATUS with IQ – escaping from the Boolean straitjacket. *Program*, Vol. 22, pp. 32-43.

PARALOG (1990) *A Guide for TRIP Managers, Version 2.4*. Stockholm: PARALOG.

Paternost, J. (1963) *The Slovenian Verbal System: Morphophonemics and Variations.* PhD Thesis, Indiana University.

Perry, S.A. and Willett, P. (1983) A review of the use of inverted files for best match searching in information retrieval systems. *Journal of Information Science*, Vol. 6, pp. 59-66.

Pogue, C. and Willett, P. (1984) An evaluation of document retrieval from serial files using the ICL Distributed Array Processor. *Online Review*, Vol. 8, pp. 569-584.

Pollitt, A.S. (1986) *An expert system approach to document retrieval: a summary of the CANSEARCH Research Project.* Technical Report Series (86/6): Huddersfield Polytechnic.

Pollock, J.J. and Zamora, A. (1984) Automatic spelling correction in scientific and scholarly text. *Communications of the ACM*, Vol. 27, pp. 358-368.

Popovič, M. and Willett, P. (1990) Processing of documents and queries in a Slovene language free text retrieval system. *Literary and Linguistic Computing*, Vol. 5, pp. 182-190.

Porter, M.F. (1980) An algorithm for suffix stripping. *Program*, Vol. 14, pp. 130-137.

Porter, M.F. (1982) Implementing a probabilistic retrieval system. *Information Technology: Research and Development*, Vol. 1, pp. 131-156.

Porter, M.F. (1983) Information retrieval at the Sedgwick Museum. *Information Technology: Research and Development*, Vol. 2, pp. 169-186.

Porter, M.F. and Galpin, V. (1988) Relevance feedback in a public access catalogue

for a research library: Muscat at the Scott Polar Research Institute. *Program*, Vol. 22, pp. 1-20.

Research Community of Slovenia (1989) *Sistem znanstvenega in tehničnega informiranja v Sloveniji*. Ljubljana: Raziskovalna skupnost Slovenije.

Robertson, S.E. (1981) The methodology of information retrieval experiment. *In:* Sparck Jones, K. (*ed.*) *Information Retrieval Experiment.* London: Butterworths, pp. 9-31.

Robertson, S.E. (1986) On relevance weight estimation and query expansion. *Journal of Documentation*, Vol. 42, pp. 182-188.

Robertson, S.E. (1990) On sample sizes for non-matched-pair IR experiments. *Information Processing and Management*, Vol. 26, pp. 739-753.

Robertson, S.E. and Sparck Jones, K. (1976) Relevance weighting of search terms. *Journal of the American Society for Information Science*, Vol. 27, pp. 129-146.

Rolland-Thomas, P. and Mercure, G. (1989) Subject access in a bilingual online catalogue. *Cataloguing and Classification Quarterly*, Vol. 10, pp. 141-150.

Sager, J.C. *et al.* (1982) Thesaurus integration in the social sciences. Part III: Guidelines for the integration of thesauri. *International Classification*, Vol. 9, pp. 64-70.

Salton, G. (1969) Automatic processing of foreign language documents. *In:* G. Salton (*ed.*) *Information Storage and Retrieval.* Report ISR-16 to the National Science Foundation, Department of Computer Science, Cornell University, Itaca, New York, pp. IV/1-IV/30.

Salton, G. (1971) *The SMART Retrieval System—Experiments In Automatic Document Processing.* Englewood Cliffs, N.J.: Prentice Hall.

Salton, G. (1975) *Dynamic Information and Library Processing.* Englewood Cliffs: Prentice-Hall.

Salton, G. (1986) Recent trends in automatic information retrieval. *Proceedings of the Ninth International Conference on Research and Development in Information Retrieval.* Washington: ACM, pp. 1-10.

Salton, G. (1988) Thoughts about modern retrieval technologies. *Information Services and Use*, Vol. 8, pp. 107-113

Salton, G. and McGill, M.J. (1983) *Introduction to Modern Information Retrieval.* New York: McGraw-Hill.

Salton, G. *et al.* (1975) A theory of term importance in automatic text analysis. *Journal of the American Society of Information Science*, Vol. 26, pp. 33-44.

Salton, G. *et al.* (1983) Extended Boolean information retrieval. *Communications of the ACM*, Vol. 26, pp. 1022-1036

Siegel, S. and Castellan, N.J. (1988) *Nonparametric Statistics for the Behavioral Sciences.* New York: McGraw-Hill.

Smeaton, A.F. (1990) Natural language processing and information retrieval. *Infor-*

mation Processing and Management, Vol. 26, pp. 19-20.

Sparck Jones, K. (1972) A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, Vol. 28, pp. 11-21.

Sparck Jones, K. (*ed.*) (1981) *Information Retrieval Experiment.* London: Butterworths.

Sparck Jones, K. and Bates, R.G. (1977) *Report on a Design Study for the "Ideal" Information Retrieval Test Collection.* British Library R&DD Report No. 5428.

Sparck Jones, K. and Tait, J.I. (1984) Automatic search term variant generation. *Journal of Documentation*, Vol. 40, pp. 50-66.

Stibic, V. (1980) Influence of unlimited ranking on practical online search strategy. *Online Review*, Vol. 4, pp. 273-278.

Tague, J.M. (1981) The pragmatics of information retrieval experimentation. *In:* Sparck Jones, K. (*ed.*) *Information Retrieval Experiment.* London: Butterworths, pp. 59-102.

Tancig, P. (1985) *Računalniško razumevanje slovenskega jezika.* PhD thesis, University of Ljubljana.

Tarry, B.D. (1978) *Automatic Suffix Generation and Word Segmentation for Information Retrieval.* M.Sc. thesis, University of Sheffield.

Tenopir, C. (1984) Full text databases. *Annual Review of Information Science and Technology*, Vol. 19. New York: Elsevier Science Publishers, pp. 215-246.

Tollefson, J.W. (1981) *The Language Situation and Language Policy in Slovenia.* Washington: University Press of Slovenia.

Toporišič, J. (1975) Main characteristics of the Slovene Language. *In:* Komac, D. and Škerlj, R. *English-Slovene and Slovene-English Dictionary.* Ljubljana: Cankarjeva založba, pp. 417-435.

Toporišič, J. (1984) *Slovenska slovnica.* 2nd ed., Maribor: Obzorja.

Ulmschneider, J.E. and Doszkocs, T. (1983) A practical stemming algorithm for online search assistance. *Online Review*, Vol. 7, pp. 301-318.

van Rijsbergen, C.J. (1979) *Information Retrieval. 2nd ed.*, London: Butterworths.

Vickery, A. *et al.* (1987) A reference and referral system using expert system techniques. *Journal of Documentation*, Vol. 43, pp. 1-23.

Vidovič-Muha, A. (1988) *Slovensko skladenjsko besedotvorje ob primerih zloženk.* Ljubljana: Partizanska knjiga.

Wade, S.J. and Willett, P. (1988). INSTRUCT: a teaching package for experimental methods in information retrieval. Part 3. Browsing, clustering and query expansion. *Program*, Vol. 22, pp. 44-61.

Wade, S.J. *et al.* (1988) A comparison of knowledge-based and statistically-based techniques for reference retrieval. *Online Review*, Vol. 12, pp. 91-108.

Wade, S.J. *et al.* (1989) SIBRIS: the Sandwich Interactive Browsing and Ranking Information System. *Journal of Information Science*, Vol. 15, pp. 249-260.

Walker, S. and Jones, R.M. (1987) *Improving Subject Retrieval in Online Catalogues: 1. Stemming, Automatic Spelling Correction and Cross-Reference Tables.* London: British Library. (British Library Research Paper 24).

Wenzel, F. (1980) Semantische Eingrenzung im Freitext-Retrieval auf der Basis morphologischer Segmentierungen. *Nachrichten für Dokumentation*, Vol. 31, pp. 29-35.

Willett, P. (1981) A fast procedure for the calculation of similarity coefficients in automatic classification. *Information Processing and Management*, Vol. 17, pp. 53-60.

Willett, P. (1985) Use of ranking methods in searches of textual and structural data bases. *Proceedings of the Ninth International Online Information Meeting.* Oxford: Learned Information, pp. 343-353.

Willett, P. (*ed*). (1988a) *Document Retrieval Systems.* London: Taylor Graham.

Willett, P. (1988b) Recent trends in a hierarchic document clustering: a critical review. *Information Processing and Management*, Vol. 24, pp. 577-597.

Willett, P. and Wood, F.E. (1989) Use of the INSTRUCT text retrieval program at the Department of Information Studies, University of Sheffield. *Education for Information*, Vol. 7, pp. 133-141.

Williams, M.E. (1985) Electronic databases. *Science*, Vol. 228, pp. 445-456.

Wolpert, S.A. (1983) A command language for the executive. *Information Services and Use*, Vol. 3, pp. 261-272.

Wood, F.E. (1981) Online teaching aids from the Department of Information Studies, University of Sheffield. *Online Review*, Vol. 5, pp. 487-494.

Wood, F.E. (1984) Teaching online information retrieval in United Kingdom library schools. *Journal of the American Society for Information Science*, Vol. 35, pp. 53-55.

Zipf, H.P. (1965) *Human Behavior and the Principle of Least Effort. 2nd ed.*, New York: Hafner Publishing Company.