Examples of Corpus Data Visualization: Collocations in Chinese

Ľuboš GAJDOŠ
Comenius University in Bratislava, Slovakia
lubos.gajdos@uniba.sk

Elena GAJDOŠOVÁ
Comenius University in Bratislava, Slovakia
gajdosova137@uniba.sk

Abstract

The article aims to show a practical procedure that can be used in the visualization of language data. The paper freely follows our previous articles about the visualization of language data in language pedagogy. We demonstrate how to retrieve language data – in our case from corpora, how to edit data in a spreadsheet program, and then in the last step, how to visualize it on the example of Legal Chinese and partly Legal German. The Javascript library Vis.js via Pyvis is chosen for the visualization of the language data.

Keywords: visualization, corpus, Chinese, Javascript library Vis.js, Python

Povzetek

Namen članka je predstaviti postopek vizualizacije jezikovnih podatkov. S tem se navezujemo na dosedanje prispevke o vizualizaciji jezikovnih podatkov pri poučevanju jezika. V članku najprej prikažemo, kako poteka pridobivanje jezikovnih podatkov, kar so v našem primeru korpusi besedil. Nato prikažemo, kako podatke urejamo z orodjem za delo s preglednicami, nazadnje pa se osredotočimo na vizualizacijo podatkov, za kar smo uporabili primer pravne kitajščine in deloma pravne nemščine. Za vizualizacijo jezikovnih podatkov smo uporabili dinamično knjižnico Vis.js in modul Pyvis v Pythonu.

Ključne besede: vizualizacija, korpus, kitajščina, dinamična knjižnica Vis.js, Python

Acta Linguistica Asiatica, 12(2), 2022.
ISSN: 2232-3317, http://revije.ff.uni-lj.si/ala/
DOI: 10.4312/ala.12.2.9-26

1 Introduction

Digitized language data can be obtained from several sources. In this article, we chose a corpus as a source of data, though data can also be obtained relatively easily from other sources. A corpus or corpus linguistics brings some benefits when compared to other sources.[1] Let us mention a few major benefits: (1) language data are already preprocessed, which means that texts are cleaned, (2) very often texts are tokenized (divided into tokens), (3) texts or tokens are annotated, and (4) corpora are equipped with statistical tools.

2 Methods

As with the data sources, there are many ways to retrieve data from a corpus in general.[2] In this chapter, only a basic procedure is introduced. Followed by the preliminary stage of the visualization – data retrieving and data editing, the visualization of data via Vis.js[3] library[4] is also shown.

2.1 Retrieving data

The most conventional way to obtain data from a corpus is to use a built-in corpus manager and, for example, Corpus Query Language (hereafter CQL). As our goal is to find the most "common" collocations in a corpus, there are probably other, more appropriate, or perhaps easier ways to tackle this task.

As confirmed by our previous research on the identification of collocations, language data which are obtained directly from the built-in functionalities offered by the NoSketch Engine is advantageous. Both corpora – the subcorpus *Zh-law* of the Hanku[5] corpus and the corpus COLEGE,[6] which serve as the data source, use the NoSketch Engine. Because there are many queries for the identification of collocations (only some of the parameters change), it is convenient to use scripting languages. In

[1] Corpus linguistics is a well-established part of linguistics. Despite some existing methodology issues, results from corpus linguistics have already proven that some areas of linguistics and language pedagogy are unimaginable without it, such as lexicography or quantitative analyses, just to mention a few. For more details on using the corpus data in language pedagogy, see Petrovčič (2022, pp. 43–47).

[2] See Gajdoš (2022b) or Gajdoš (2020) for details.

[3] Vis.js is a dynamic, browser-based visualization (JavaScript) library (*Vis.js*, n.d.).

[4] As the data retrieving script is written in Python, Pyvis library should be used to produce Javascript code. Pyvis is designed as a wrapper around the popular Javascript Vis.js library (*Pyvis*, n.d.).

[5] The Hanku corpus is a corpus of the Chinese Language. The zh-law is a subcorpus of legal Chinese (Gajdoš et al., 2016).

[6] The COLEGE is a corpus of legal German. See Gajdošová & Gajdoš (2018) for details.

our case, by retrieving language data from corpora, programming languages like Python[7] and the JSON format[8] are used, however, the following procedure describes a "manual" way, and it may also be used.

1. Get keywords (hereafter KWIC)[9] from the corpus by using query, e.g. CQL:[10] [tag="VV|NN"], then use node forms functionality, and finally save as .txt format.[11]
2. Again, based on our previous research, it is advantageous to look for collocations on the right and left sides separately.[12] LogDice score is chosen as the basic criterion (the measure of association), and the span is set to 5 to the left and 5 to the right side from KWIC for Chinese, and 10 to the left and 10 to the right for German.[13] The results are sorted by node forms and frequency for Chinese, and then lemma for German. The results are saved as a .txt file. As spreadsheet programs may not work properly with .txt formats, it is advisable to simply change the .txt filename extension (a suffix to the name) to .csv.[14]

2.2 Data editing

In this step, the data will get equipped with parameters and will further be formed in the way to best suit the visualization. For our purposes, the Pandas[15] library was used to modify the data. The following steps show some manual data modifications.

3. In a spreadsheet program,[16] the results from both sides are merged into a single file based on the LogDice score maintaining the side designation.[17]

[7] Python (version 3) is a programming language, for more information see *Python* (n.d.).

[8] JSON (JavaScript Object Notation) is a lightweight data-interchange format, for more information see *JSON* (n.d.).

[9] Keywords can be selected based on one's own requirements, e.g. only verbs and nous, as it is in our case (tags VV or NN for Chinese).

[10] The CQL query means "searching for all verbs VV or (symbol |) nouns NN". In the next step, the functionality *Node forms* sorts tokens by frequency.

[11] The most frequent tokens in a corpus do not necessarily create the strongest collocations (measured by the Logdice score).

[12] As will be shown in the following chapters, this information is used in the visualization.

[13] See Gajdošová (2022) or Gajdoš (2022a) for more information.

[14] Abbreviation CSV stands for "Comma-Separated Values". See *CSV* (n.d.) for details.

[15] See *PANDAS* (n.d.) for more details.

[16] Here it is worth noting that not all spreadsheet programs work properly with the CSV format. Libreoffice Calc has proven to be a good solution in this regard. For more details, see https://www.libreoffice.org/discover/calc/.

[17] The side designation may be added manually after obtaining data from a corpus for each side separately.

4. The previous steps are repeated for all the keywords. The results are then combined into one single .csv file (sorted by the LogDice score). This can be considered as the basis for visualization.

Table 1 below shows the results for Legal Chinese. POS tags of KWIC (column POS_kwic) and collocators (POS_item) are added manually at the end of the entire search. According to the requirements, it is possible to add other information that can be obtained from the corpus, such as the author's gender, period of origin, and others.

Table 1: Identified collocations in the corpus *Zh-law*

KWIC	Side	Item	Logdice	Corpus	POS_kwic	POS_item
部门	LS	主管	12,921	Zh-law	NN	NN
人民	RS	政府	12,914	Zh-law	NN	NN
不	RS	得	12,856	Zh-law	AD	VV
人民	LS	中华	12,803	Zh-law	NN	NR
人民	RS	共和国	12,789	Zh-law	NN	NN
部门	LS	行政	12,483	Zh-law	NN	NN
行政	RS	部门	12,455	Zh-law	NN	NN
人民	RS	法院	12,183	Zh-law	NN	NN
规定	LS	本	12,116	Zh-law	NN	DT

2.3 Visualization

Data visualization can be done in different ways. We here demonstrate only one of the possible ways of using Vis.js – network. The network in the Vis.js library mainly consists of nodes and edges. In our case, the keywords (KWIC) and identified collocators (item) are chosen as the nodes. The following code shows how to create a simple network with three nodes and two connections (edges) as an arrow based on the side (LS).

```
// create an array with nodes
var nodes = new vis.DataSet([
  { id: "部门", label: "部门" , shape: "dot", value: 2, group: 1},
  { id: "主管", label: "主管" , shape: "dot", value: 1, group: 2},
  { id: "行政", label: "行政" , shape: "dot", value: 1, group: 3},
]);
// create an array with edges
var edges = new vis.DataSet([
  { from: "主管", to: "部门" },
  { from: "行政", to: "部门" },
]);
var edges = [{ from: "部门", to: "主管", arrows: "from" },
{ from: "部门", to: "行政", arrows: "from" },
];
```

As can be seen from the code above, there are many parameters that can be used for nodes and edges. In our visualization, the following parameters are used – shape (dot), value (based on the number of connections/edges to nodes), and group (based on POS tag). Nodes can be dragged via a left mouse click. Figure 1 shows the visualization of a given code.

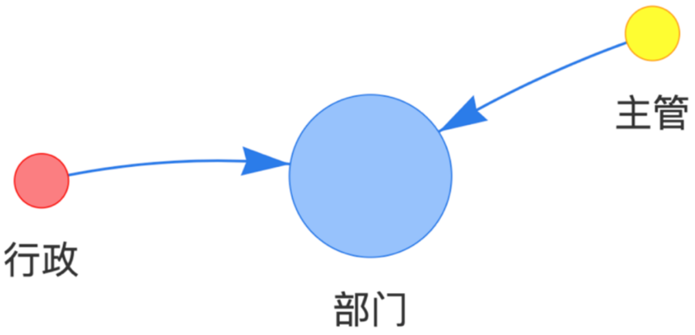


Figure 1: Example of creating a simple network in Vis.js

It is possible to create nodes and edges manually, yet with a large amount of data, this procedure is very laborious and time-consuming. For these reasons, it is very convenient to use one of the scripting languages, e.g. Python.[18]

[18] In the first step, the nodes are added within a for loop. In Python with Pandas are used, i.e. “for index, row in df.iterrows():”. For more details, see https://pandas.pydata.org (*PANDAS*, n.d.).

The node ID is then the KWIC or item token. One KWIC or item (any token) is only one node. Edges are connected according to the side. The value of the node (the size) is based on the number of edges to other nodes. The result of the visualization is a .html file that may be displayed in a web browser such as Firefox, Chrome, or any other.

Because displaying a large amount of nodes (more than 1000) is computationally intensive, it is a good idea to choose a compromise number of nodes that are meaningful. It is appropriate to take this step when modifying/editing the data. Alternatively, select only some keywords (such as nouns and verbs) and then display them separately.

3 Practical use of visualization

Before showing a practical use of the visualization, is necessary to draw attention to the limits of the visualization:

- since the tagsets for corpora are different, one must expect a different proportion of POS tags in different corpora
- in the case of using the German corpus Colege, it is necessary to consider the relatively small size of the corpus, which may affect the result from the statistical point of view
- the visualization is only as relevant as the obtained data
- when comparing more languages, it is important to pay attention to genetic and typological differences between them.

3.1 Strongest collocations in Legal Chinese

The following example demonstrates possibilities of visualization in the Vis.js library. As already demonstrated, searching for the strongest collocation to one token (KWIC) in Legal Chinese, measured by the Logdice statistical measure, is quite simple. A recursive algorithm is used to search for the strongest collocations in the corpus.[19] The following POS tags are excluded from the search: punctuation (PU), time nouns (NT), numbers (CD, OD), sentence particles (SP) and all markers DE 的, 得, 地 (D.*).[20] Table 2 shows only a small portion of the result.[21] The whole table has 935 rows.

[19] The searching function is called itself until the Logdice value is below a certain value. For more details about recursion, see https://openbookproject.net/thinkcs/python/english3e/recursion.html

[20] Needless to say, these restrictions may be set arbitrary, and, in this case, these restrictions are applied to KWIC and collocators (Item).

[21] The very first result is particularly interesting. The formula for calculating Logdice shows that the maximum theoretical value is 14, yet this result is higher (Rýchly, 2008). The explanation is in this case quite simple – cooccurrence count (689) of tokens *yǒuqī* 有期 (period) *túxíng* 徒刑

Table 2: Strongest collocations in Legal Chinese

KWIC	Side	Item	Logdice	Corpus	POS_kwic	POS_item
徒刑	LS	有期	14,022	Zh-law	NN	JJ
中华	RS	共和国	13,979	Zh-law	NR	NN
共和国	LS	中华	13,979	Zh-law	NN	NR
自治区	RS	直辖市	13,895	Zh-law	NN	NN
直辖市	LS	自治区	13,895	Zh-law	NN	NN
徇私	RS	舞弊	13,894	Zh-law	VV	NN
舞弊	LS	徇私	13,894	Zh-law	NN	VV
款罪	LS	犯前	13,882	Zh-law	NN	VV
犯前	RS	款罪	13,882	Zh-law	VV	NN
有期	RS	徒刑	13,861	Zh-law	JJ	NN
自治区	LS	省	13,779	Zh-law	NN	NN
省	RS	自治区	13,779	Zh-law	NN	NN
省	RS	直辖市	13,733	Zh-law	NN	NN
直辖市	LS	省	13,733	Zh-law	NN	NN
交通管	RS	制员	13,601	Zh-law	NN	NN
刑事	LS	追究	13,596	Zh-law	NN	VV
追究	RS	刑事	13,596	Zh-law	VV	NN
撤职	LS	降级	13,484	Zh-law	VV	VV
降级	RS	撤职	13,484	Zh-law	VV	VV

(imprisonment; fixed-term imprisonment) is higher than candidate count (619) of 有期. This phenomenon may be caused by errors in tokenization.

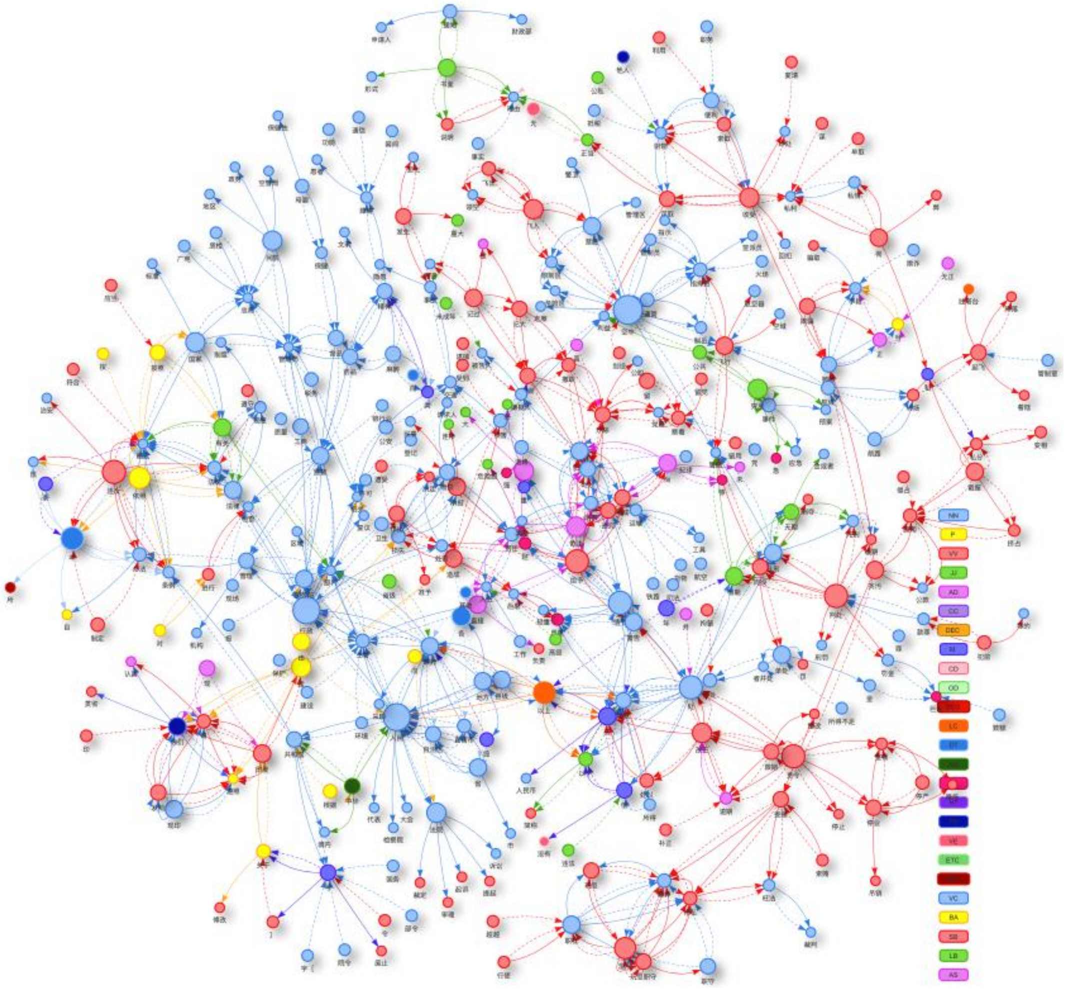


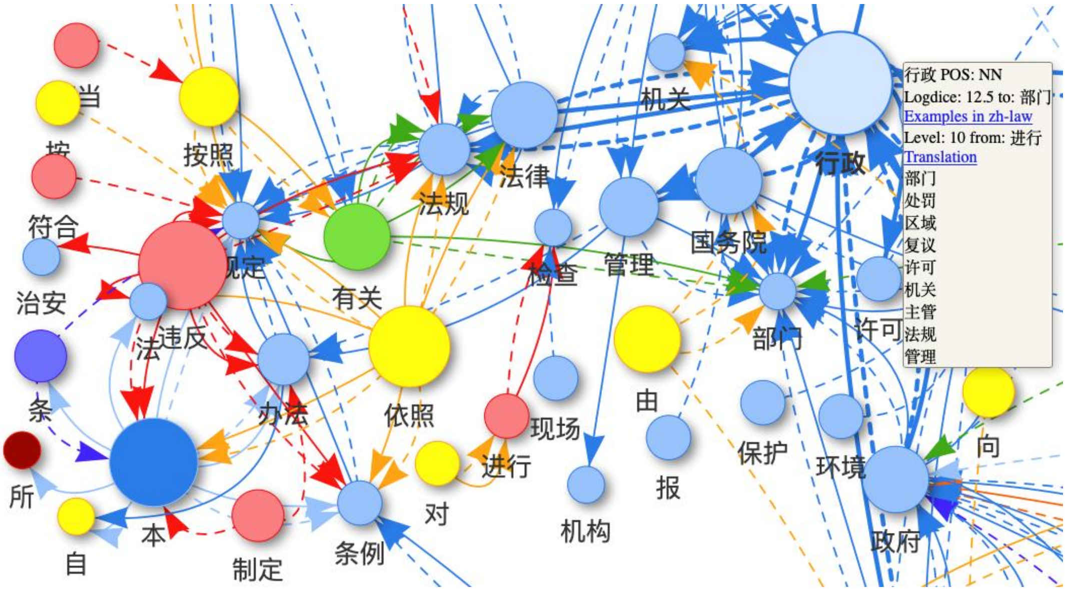
Figure 2: Network of 935 strongest collocations in Legal Chinese

As the printed version of the visualization is fairly limited, possibilities of the visualization are demonstrated in the multiple figures. Figure 3 shows the following:

- there are nodes (tokens) that make more connections than others, the size of a node reflects this information (mostly blue are nouns and red nodes are verbs)
- some parts of speech are more common than others, e.g. blue nodes (nouns)[22]
- clusters of blue nodes (nouns) and red (verbs) are the most common
- arrows point to the collocators (dashed edges to the collocators on the right side, solid-line edges to the collocators on the right side) in the word order.

[22] It is worth noting that due to polysemy or conversion (zero derivation), POS tags are not always adequate in a certain collocation in Chinese.

Though the above information is also available in the .csv file, we believe that their visualization is more easily readable to a student or in the field of second language acquisition in general. This is also because some information, as shown in Figure 3, can only be retrieved from the table via a search, while in the figure, this information is available by clicking on a node. Also, the figure can be zoomed in or out using the mouse.



行政 POS: NN
Logdice: 12.5 to: 部门
Examples in zh-law
Level: 10 from: 进行
Translation
部门
处罚
区域
复议
许可
机关
主管
法规
管理

Figure 3: Zoom in and viewing connection to other tokens

After clicking on a node, the pop-up menu would show that the node *xíngzhèng* 行政 (administrative) is a noun (NN), which makes collocations with the following tokens in the column: *bùmén* 部门 (department), *chǔfá* 处罚 (punish), and others. At the same time, the edges are also highlighted. Furthermore, by clicking on the connected node, one can obtain information on the collocators to the connected node as well. The pop-up menu in Vis.js also offers other options, such as a hyperlink directly to examples in the corpus, to the web translator, statistical data, and others.

3.2 Comparison of synonyms

There are synonyms in every natural language and one of the difficulties in translating them or in the L2 acquisition is to find their equivalences. Our empirical experiences show that it is appropriate to seek an equivalence, not at the level of words (tokens) but at least at the level of collocations (bigrams, n-grams), as has already been shown by Benická (2017), and others.[23]

[23] For more details about translation issues, see Benická (2017).

Let us start with an example of the following three Chinese synonyms: "according to" *ànzhào* 按照，*gēnjù* 根据，*yīzhào* 依照. Table 3 shows the first 10 rows. The whole table consists of 193 rows.

Table 3: Collocations to given prepositions

KWIC	Side	Item	Logdice	Corpus	POS_kwic	POS_item
按照	RS	规定	11,477	Zh-law	P	NN
依照	RS	法	11,02	Zh-law	P	NN
根据	RS	需要	11,007	Zh-law	P	VV
依照	RS	法律	10,958	Zh-law	P	NN
按照	RS	有关	10,86	Zh-law	P	JJ
按照	LS	应当	10,809	Zh-law	P	VV
根据	RS	共和国	10,781	Zh-law	P	NN
根据	RS	中华	10,78	Zh-law	P	NR
依照	RS	规定	10,769	Zh-law	P	NN
依照	RS	条例	10,685	Zh-law	P	NN

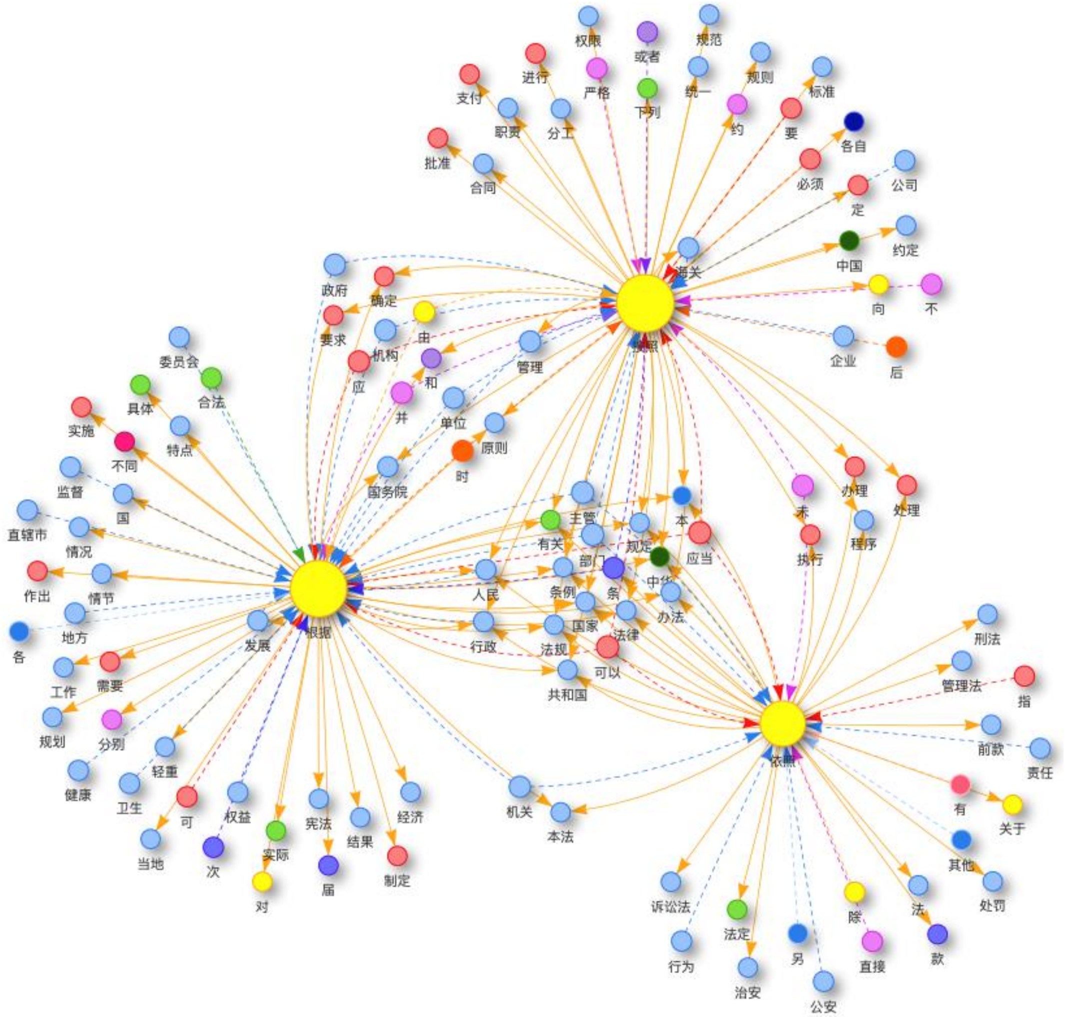


Figure 4: Collocators to prepositions *ànzhào* 按照, *gēnjù* 根据 and *yīzhào* 依照

As can be seen from Figure 4, these prepositions have mutual collocators, of which some are typical for two prepositions and some even for one preposition only. Let us zoom in to see more details.

On the left side of Figure 5, there is a group of *gēnjù* 根据 and *ànzhào* 按照 collocators. The group in the middle are collocators of all three prepositions. The group on the right side are the mutual collocators to *ànzhào* 按照 and *yīzhào* 依照.

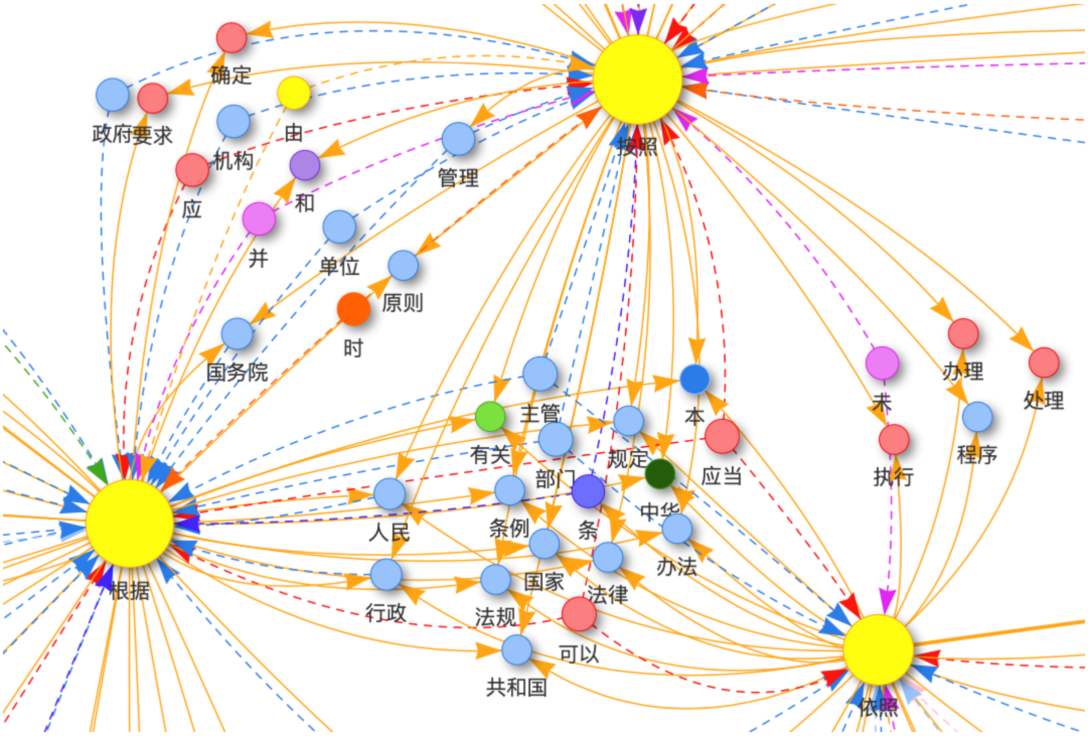


Figure 5: Mutual collocators of the prepositions

3.3 Comparison of modal verbs in legal texts

In the case of translating legal texts, for example, it is very important to find an equivalence in translated languages, in our case for modal verbs. Due to the ongoing research,[24] the language pair Chinese – German is chosen as an example.

There are no specific tags for modal verbs[25] in the Chinese corpus, so modal verbs must be selected manually as follows by the CQL query:

> [word=”要|能|会|可以|可|想|能够|需要|应|可能|必须|得|应该|需|应当|该|必|想要|愿意|能否|愿|须|得以” & tag=”VV”]

Table 4 below shows the first 10 collocations only. The whole table has 984 rows.

[24] For more details, see Gajdošová (2022).

[25] Identifying modal verbs is not a trivial operation, since many modal verbs in Chinese may also be e.g. transitive verbs.

Table 4: Collocations of modal verbs

KWIC	Side	Item	Logdice	Corpus	POS_kwic	POS_item
得	LS	不	12,846	Zh-law	VV	AD
会	LS	保监	11,961	Zh-law	VV	VV
会	RS	同	11,601	Zh-law	VV	DT
会	LS	银监	11,55	Zh-law	VV	NN
必	RS	修课	11,415	Zh-law	VV	VV
可	LS	行政许	11,366	Zh-law	VV	NN
需	LS	确	11,342	Zh-law	VV	AD
可能	RS	影响	11,134	Zh-law	VV	NN
能	LS	只	11,068	Zh-law	VV	AD
可以	LS	也	11,018	Zh-law	VV	AD

As can be seen from Figure 6 below, some modal verbs collocate more often than others.

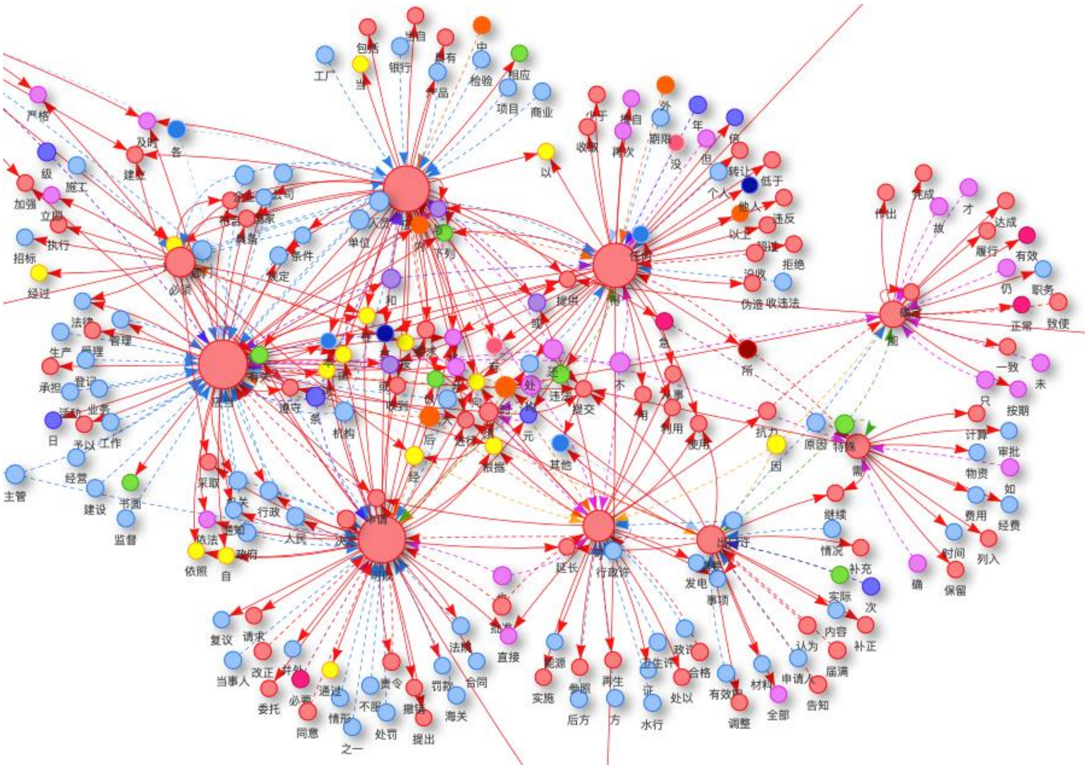


Figure 6: Collocability of modal verbs in Chinese

From the list of modal verbs, the token that collocates the most is the verb *yīngdāng* 应当 (should; must), *yīng* 应 (should), *kěyǐ* 可以 (may), *dé* 得 (must), and may express deontic modality. The collocability of other verbs is rather limited or they do not collocate at all. Such are *yuànyì* 愿意 (willing to), *yuàn* 愿 (willing to), *yīnggāi* 应该 (should), and others.

Let us zoom in. It is obvious that the modal verbs *yīngdāng* 应当 (should; must), *yīng* 应 (should), and *dé* 得 (must) have mutual collocators. On the other hand, some modal verbs such as *yīnggāi* 应该 (should) and *gāi* 该 (should) rarely collocate in Legal Chinese, if at all.

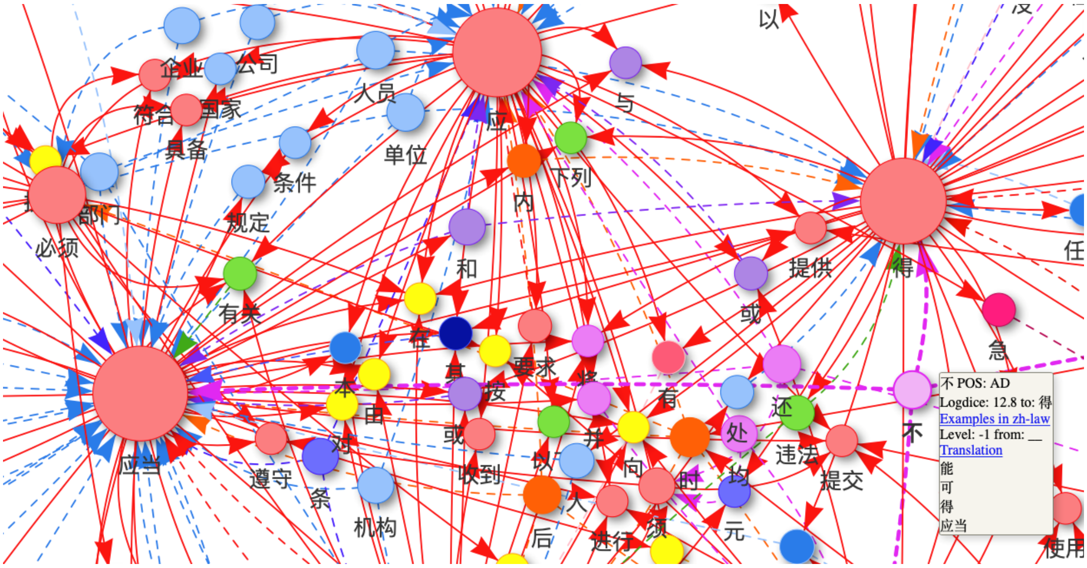
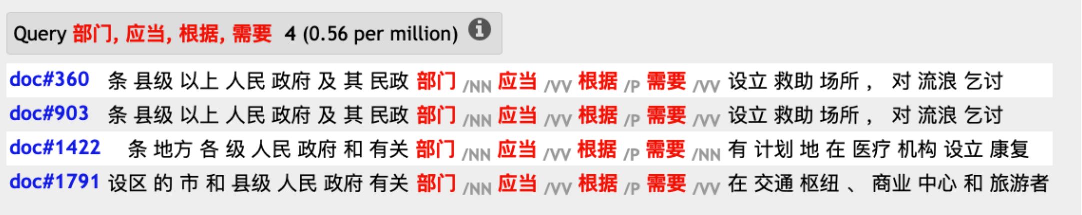


Figure 7: Collocability of verbs (from left) *yīngdāng* 应当, *yīng* 应 and *dé* 得

Now let us look at a practical example of using the network in L2 acquisition and select the node *bùmén* 部门 (department; division). This node points to the verb *yīngdāng* 应当 (must) and this verb, among others, points to the preposition *gēnjù* 根据 (according to), which further points to the verb *xūyào* 需要 (a need; to need). Searching for the above tokens in the Zh-law subcorpus brings the results presented in Figure 8.



Query 部门, 应当, 根据, 需要 4 (0.56 per million)

doc#360 条 县级 以上 人民 政府 及 其 民政 部门 /NN 应当 /VV 根据 /P 需要 /VV 设立 救助 场所 ， 对 流浪 乞讨

doc#903 条 县级 以上 人民 政府 及 其 民政 部门 /NN 应当 /VV 根据 /P 需要 /VV 设立 救助 场所 ， 对 流浪 乞讨

doc#1422 条 地方 各 级 人民 政府 和 有关 部门 /NN 应当 /VV 根据 /P 需要 /NN 有 计划 地 在 医疗 机构 设立 康复

doc#1791 设区 的 市 和 县级 人民 政府 有关 部门 /NN 应当 /VV 根据 /P 需要 /VV 在 交通 枢纽 、 商业 中心 和 旅游者

Figure 8: Example of the n-gram in Legal Chinese

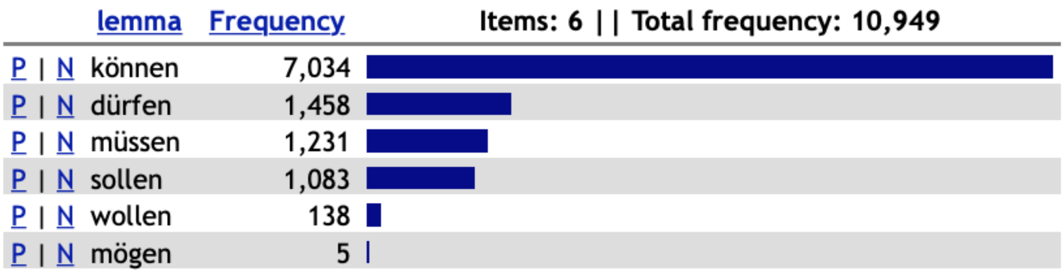
The visualization may not appear clear enough at first sight. If so, it is advantageous to reduce the number of collocators or modify the parameters of the physical model, e.g. change the value of constants (gravitationalConstant), extend the length of the edges (springLength), etc. though these are rather limitations resulting from the printed version. The .html version solves these problems partially or completely.

The downside of the visualization, as the parameters are set in this example (node size based on the number of collocators), is also the inability to display collocation strength based on the Logdice score. An example in this case is the negation *bù* 不 (not) which makes the strongest collocation of all modal verbs, in this case with *bù dé* 不得 (must not). Besides, the negation *bù* 不 can also collocate with other modal verbs in Legal Chinese.

Now, let us compare the above situation with the one in Legal German. Taking the same example, we would like to illustrate a different approach to the same problem in two different corpora and languages. This is not an exhaustive analysis of the issue of equivalence in these languages but just a sample of the possibilities.

As for the modal verbs in German, there is a dedicated tag for modal verbs in the Colege corpus and it may be identified very easily by the CQL:[26] [tag="VMFIN| VMINF"]

Table 5: Modal verbs in Legal German



Items: 6 | | Total frequency: 10,949

	lemma	Frequency	
P | N	können	7,034	████████████████████
P | N	dürfen	1,458	████
P | N	müssen	1,231	███
P | N	sollen	1,083	███
P | N	wollen	138	▌
P | N	mögen	5	▏

[26] The VMFIN tag means a finite form of a modal verb, the symbol "|" OR and VMINF means an infinitive form of a modal verb.

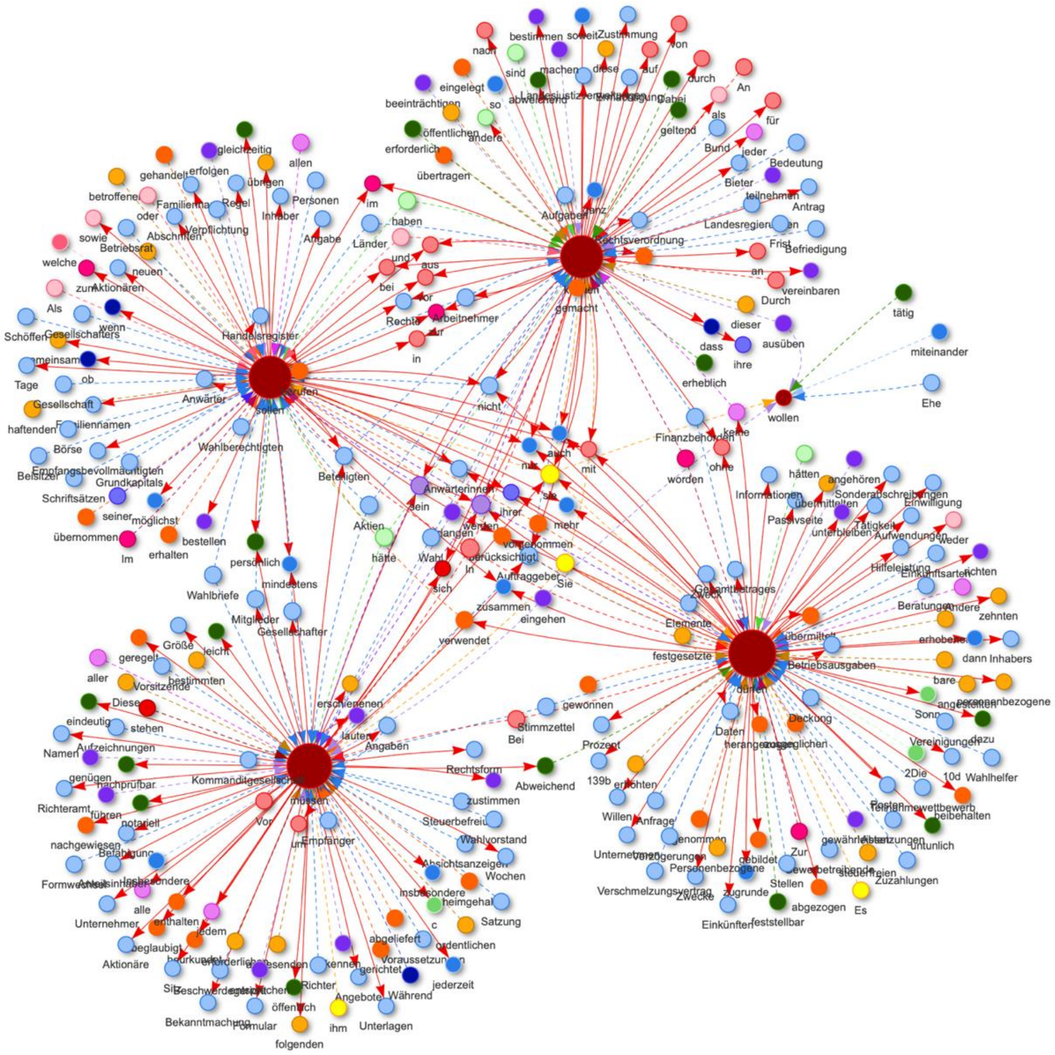


Figure 9: Modal verbs in Legal German

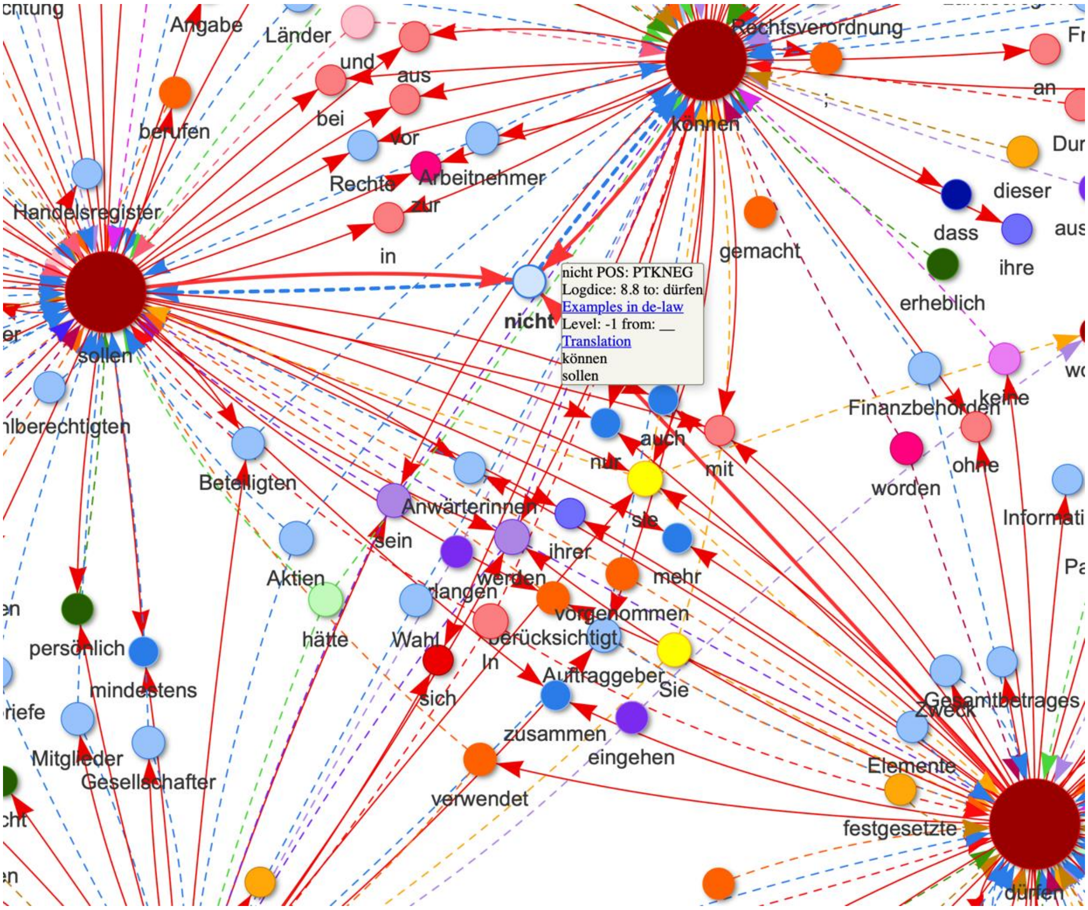


Figure 10: Modal verbs with negations in Legal German

As can be seen from the visualization, the modal verbs *sollen* (should, shall), *dürfen* (may), *können* (can) can be associated directly with the negation *nicht* (not) (similar to *keine* none), but the verb *müssen* (must) does not take a negation. Therefore, when comparing the situation in Legal German vs. Legal Chinese, it is clear that the collocational preferences at the negation are different.

4 Conclusion

In this article, we have briefly demonstrated the whole process of a language data visualization – from retrieving, via data editing to the visualization. We have shown some examples of the visualization.

In the end, there is still one question to answer: is this kind of visualization useful? We think that this type of visualization is primarily intended for the area of language acquisition. We believe that visualization can help get a basic overview of the examined data – whether it is a view of the language register (as in our case), or to a literary work of one author, or else just to compare the collocability of individual words (tokens). With appropriately selected parameters and the number of nodes, it is possible to very clearly show typical features of a text, register, or the relation between words. To conclude, the visualization of linguistic data has its place in language data processing and sometimes provides a clearer insight into the issue.

References

Benická, J. (2017). Archaizujúci jazyk v čínštine a jeho prekladanie do slovenčiny. (Historicizing Language in Chinese and its Translation into Slovak). In D. Veverková, I. Kolečáni Lenčová, M. Ľupták & Z. Danihelová (Eds.), *Aplikované jazyky v univerzitnom kontexte IV* (pp. 58-68). Technická univerzita.

CSV. (n.d.). Retrieved March 15, 2022, from https://www.w3.org/TR/tabular-data-primer/#tabular-data

Gajdoš, Ľ., Garabík, R., & Benická, J. (2016). The New Chinese Webcorpus Hanku—Origin, Parameters, Usage. *Studia Orientalia Slovaca, 15*(2), 21-33.

Gajdoš, Ľ. (2020). Verb Collocations in Chinese – Retrieving, Visualization and Analysis of Corpus Data. *Studia Orientalia Slovaca, 19*(1), 121-138.

Gajdoš, Ľ. (2022a). Vizualizácia jazykových dát ako didaktická pomôcka na príklade korpusu čínskych právnych textov (Visualisation of Linguistic Data as a Didactic Tool on the Example of the Corpus of Legal Chinese). In *Kontexty súdneho prekladu* (pp. 7-22).

Gajdoš, Ľ. (2022b). *Praktická korpusová lingvistika – čínština* (Practical Corpus Linguistics – Chinese Language). Univerzita Komenského.

Gajdošová, E., & Gajdoš, Ľ. (2018). Korpus nemeckého právneho textu COLEGE (Corpus of Legal German). In *Kontexty súdneho prekladu a tlmočenia*, 7, 40-47.

Gajdošová, E. (2022). Korpusbasierte Analyse von Rechtstexten in slowakischer und deutscher Sprache mit besonderem Augenmerk auf Verb-Nomen-Kollokationen [Unpublished doctoral dissertation]. Univerzita Komenského.

JSON. (n.d.). Retrieved March 15, 2022, from https://www.json.org/json-en.html

PANDAS. (n.d.). Retrieved March 14, 2022, from https://pandas.pydata.org

Petrovčič, M. (2022). Chinese Idioms: Stepping Into L2 Student's Shoes. *Acta Linguistica Asiatica*, *12*(1), 37-58. https://doi.org/10.4312/ala.12.1.37-58

Python. (n.d.). Retrieved March 15, 2022, from https://www.python.org

Pyvis. (n.d.). Retrieved March 16, 2022, from https://pyvis.readthedocs.io/en/latest/#

Rýchly, P. (2008). A Lexicographer-Friendly Association Score. In P. Sojka & A. Horák (Eds.), *Recent Advances in Slavonic Natural Language Processing* (pp. 6-9). Masaryk University.

VIS.JS. (n.d.). Retrieved March 16, 2022, from https://almende.github.io/vis/docs/network/