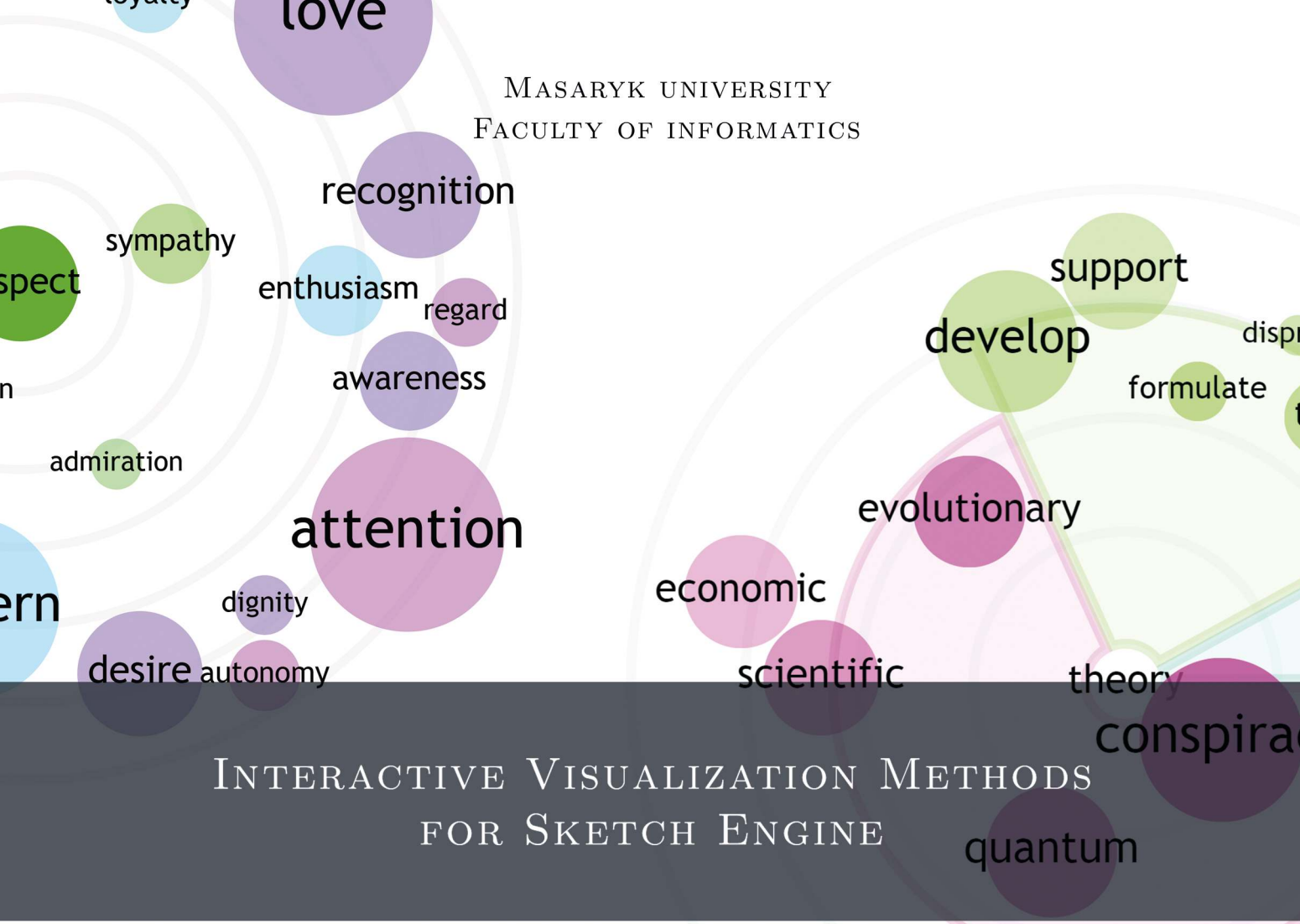
Masaryk university
Faculty of informatics



Interactive Visualization Methods for Sketch Engine

Master thesis

Lucia Kocincová

2015

Masaryk university
Faculty of informatics

Interactive Visualization Methods for Sketch Engine

Master thesis

Lucia Kocincová

2015

Declaration

I declare that this Master thesis is my original work and that I have written it independently. All sources and literature that I have used during elaboration of the thesis are correctly cited with complete reference to the corresponding sources.

Abstract

Visualization is undoubtedly one of the most desired methods for displaying data, especially when dealing with so called big data. Visualization can uncover unnoticed and hidden relationships within the data and in addition, it enables the users to understand and interpret the data with less effort.

This thesis focuses on interactive visualizations generated from the corpora data. First, it introduces the state-of-the-art tools for corpora visualizations and a corpus management system named *Sketch Engine*, for which numerous design concepts were created. Then four of them – corpora overview, thesaurus, word sketch and word sketch difference – were implemented as an online application with the main use of the *Data-Driven Documents* library. Last, these visualizations were evaluated by the user testing which revealed that the implemented concepts were not only graphically very appealing but also helpful. Therefore, the interactive visualizations will be incorporated in the *Sketch Engine* online interface in the upcoming future.

Keywords

information visualization, interactive visualization, corpora, Sketch Engine, concordance, thesaurus, word sketch, word sketch difference, Data-Driven Documents, generative design

Acknowledgment

I would like to thank my supervisor Barbora Kozlíková for her advice, enthusiasm and all the time she has spent during supervising.

Subsequently, I would like to express my gratitude to Sketch Engine developers who had spent much time discussing the concepts and explaining the algorithms behind the system, especially my consultant Vít Baisa, but also Miloš Jakubíček and Vojtěch Kovář.

Also, many thanks to all 23 people that participated in the evaluation and provided their honest feedback.

Last, but not least, I would like to thank my family and Izy for the support given not only during this project.

In the end, thanks also to *faiax12* for breaking down few years ago which lead to a series of fortunate events.

Contents

1
Introduction

Mankind is living in digital age not only because of an extensive use of technological devices but also because it produces more digital than analogical data for the first time in history (Hilbert and Lpez, 2011). As a result of this overwhelming amount of information, we are still looking for a way to automatically explore and summarize data. One of the best and demanded techniques how to present and look through data is visualization. Visualization is not only about generating nice images because evidences of its usefulness and impact on the quality of life can be found in numerous research studies.

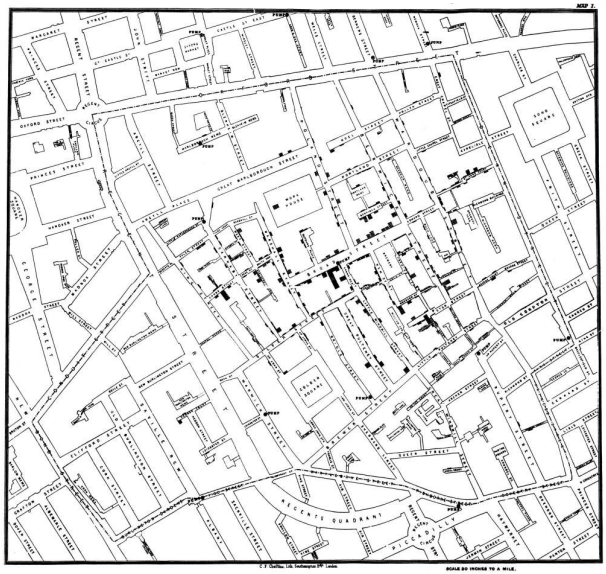


Figure 1.1: Cholera map made by John Snow (Snow, 1855)

Interactive visualization is a subset of a larger field known as information visualization, which is also sometimes referred to as informatics, that crosses the disciplinary boundaries of computer science, design, statistics, psychology, cognition, neuroscience, and the basic sciences. (Ferster, 2012) [Information visualization is the] use of computer-supported, interactive, visual representations of abstract data to amplify cognition. (Card et al., 1999)

Visualization of information is not a new method of dealing with data. Quite the contrary, one of the most known visualizations dates back to the 19th century, when a doctor John Snow indicated on a map a number of deaths caused by cholera in 1855 using simple black rectangles (Snow, 1855), as can be seen in Figure 1.1. It became clear that there were too many cases around a water pump to

consider them only a coincidence. After further investigation and closing the pump, the doctor confirmed that the disease was caused by contaminated water. At that time, this was very important discovery as it was not previously known that cholera could spread through water.

In 1973, the importance of displaying data in graphical representations was demonstrated by Francis Anscombe, an English statistician, who made four sets of data that have almost identical basic statistical properties. These data sets are denoted as the Anscombe's quartet. Each data set is made out of eleven pairs of variables (x, y) and the first three sets have even the same value x. The analysis of different data sets can be narrowed down to comparing basic statistical properties, like mean, sample variance, or correlation between variables. Anscombe's quartet has these features almost identical - all means and sample variances of variable x are exactly the same and other features are the same when rounded to 2 or 3 decimal places. Linear regression line is also almost identical as can be seen in Figure 1.2 (marked by blue colour). However, the relationships between these sets are not even nearly similar as can be immediately recognized when looking at the graphs.

Displaying relations or abrupt alterations are not the only useful matters that can be provided by visualization. Extraction of information and following abstraction is another approach when dealing with complex data or system. For example, eminent redesign of the London underground map done in 1933 by Harry Beck illustrates the way we perceive and understand information. The new schematic map shows only straight lines with 90 or 45 degree crossings as seen in Figure 1.3 (right). Decoding information from such visualization is easier than from a map that preserves geographical coordinates, because the actual length distances between places are irrelevant for a

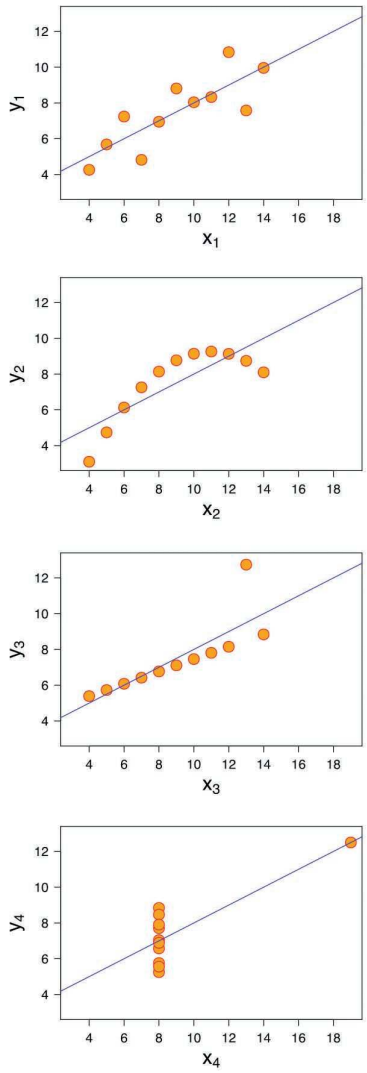
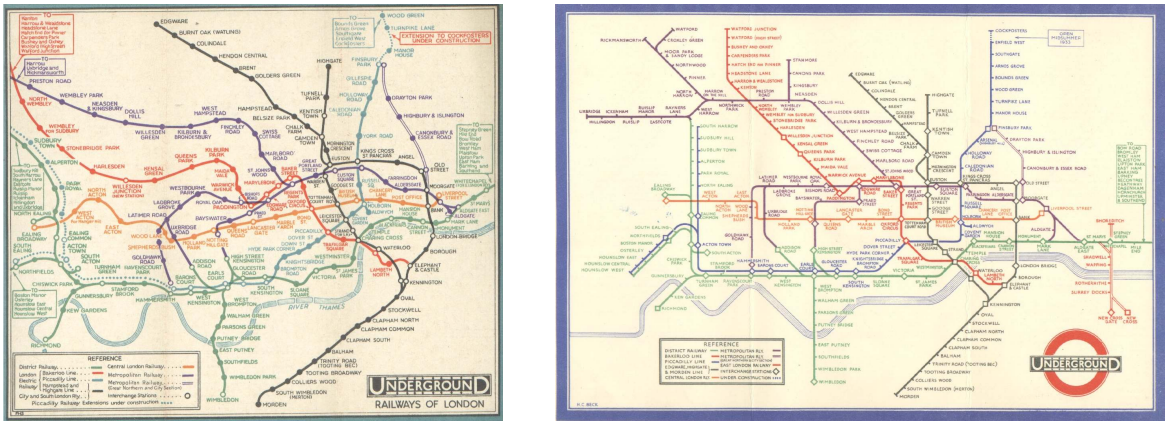


Figure 1.2: Anscombe's quartet (Wikimedia Commons, 2015)

passenger. Leaving out the geographical precision from maps of public services is since then heavily reused as schematic pictures save time and are generally more usable, which proves that graphical representation of information is an important issue even in everyday life. (Iliinsky, 2010)



(a) map from 1932 **(b)** map from 1933

Figure 1.3: London Underground Map

In the age of big data, there are numerous problems that can be inquired through visualizations. Any field of interest can have vast information that can be mapped onto graphical elements and analysed. One research field that has expanded in the last years is natural language processing, also known as NLP, whose basic research medium is a text corpus. A corpus is a collection of texts, which is machine-readable (McEnery and Wilson, 1996) and nowadays stored exclusively in an electronic form – this enables to analyze enormous amounts of data in a fraction of time and therefore to bring a whole book library or parts of the Web right to a user. For searching through corpora, a corpus query system is usually used. It also enables to analyze a given language or text using statistical features and tools.

This thesis focuses on visualizations of text corpus output from the system named *Sketch Engine*, whose functions are explained in detail in the next chapter. The main objective of this thesis is not only to analyze the state-of-the-art tools in text visualization, or to design a plentiful number of visualization methods for *Sketch Engine*, but also to implement selected concepts, so they can help users to easily understand and interpret the data.

The second chapter summarizes the state-of-the-art visualization tools for text and corpora. It provides a quick insight into the main attributes and features which were also used to analyze the advantages and limitations of the particular approaches and graphical representations.

The knowledge gained from the second chapter was used to design various visual concepts for main features of the Sketch Engine which are illustrated in the third chapter. The ideas behind the concepts and methods are mentioned as well as the decisions that led to the selection of the final concepts.

The fourth chapter consists of a short description of what technologies were considered and also the contemporary tools which were necessary during the development phase.

Details regarding implementation are stated in the fifth chapter along with the final visualization outputs.

The sixth chapter describes the purpose and methods of usability testing in which real users of the Sketch Engine participated in. Subsequently, a comprehensive evaluation from their feedback is included with conclusions regarding the visualizations.

The last chapter reviews the results and the outcomes along with the short description of future work that is planned to be accomplished even after the completion of this thesis.

2
Background

This chapter first introduces the Sketch Engine and its core features. Then it focuses on the state of the art tools in corpora visualization which are divided into seven categories depending on the main purpose of these tools.

2.1 Sketch Engine

Sketch Engine is for anyone wanting to research how words behave.[1]

[1] http://www.sketchengine.co.uk/

Sketch Engine, later mentioned only in its abbreviated form as SkE, is a corpus query system which was firstly introduced at *the Eleventh EURALEX International Congress* in 2004 (Kilgarriff et al., 2004) and it also stands for a web service which is available at http://www.sketchengine.co.uk/. Since then, it became the leading corpora tool in lexicography and offers as many as 75 different languages and 200+ text corpora (Kilgarriff et al., 2014), of which many belongs to the largest corpora ever made, as illustrated in Figure 2.1.

Figure 2.1: Selected languages in SkE and their corpora

An open source version was released as NoSketchEngine to support free and powerful management of building corpora. The tool supports also concordance querying and word list generation, but it does not provide thesaurus or word sketches. (Rychlý, 2007)

2.1.1 *Analysis of site statistics*

Statistics presented below were generated from the official website of SkE[2] and its site with beta version[3] using AWStats logfile analyzer[4]. However, some universities and companies have their own installation, therefore this data shows only partial usage of SkE.

Figure 2.2 shows the five most viewed corpora in SkE from January 2013 until April 2015, from which most used is *The British Academic Written English Corpus (BAWE)*[5]. It is a collection of almost 3,000 student assignments from different disciplinary areas. It is a corpus with an open access as well as the second most used corpus *ACL Anthology Reference Corpus*[6] which is made of publications from the field of computational linguistics. The third most used corpus is the *British National Corpus*[7] which, unlike others, has also a spoken collection of texts. The whole corpus was designed to include a wide range of British English from the late 20th century. The fourth most used corpus is the *enTenTen* which is a collection of almost 12 billion words made by Web crawling done by SkE developers.

[2] http://www.sketchengine.co.uk/

[3] beta.sketchengine.co.uk

[4] http://www.awstats.org/

[5] http://www.reading.ac.uk/internal/appling/bawe/sketch_engine_bawe.htm

[6] http://acl-arc.comp.nus.edu.sg/

[7] http://www.natcorp.ox.ac.uk/

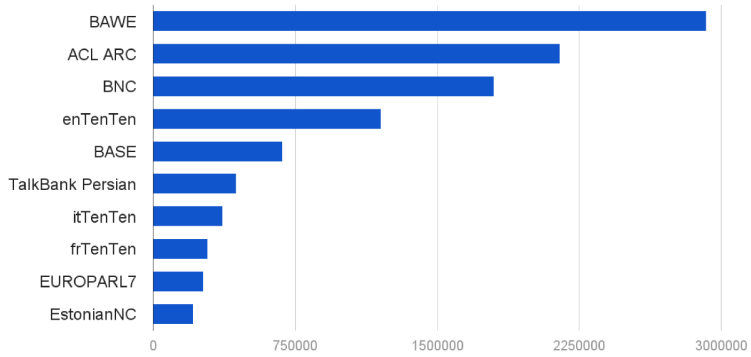
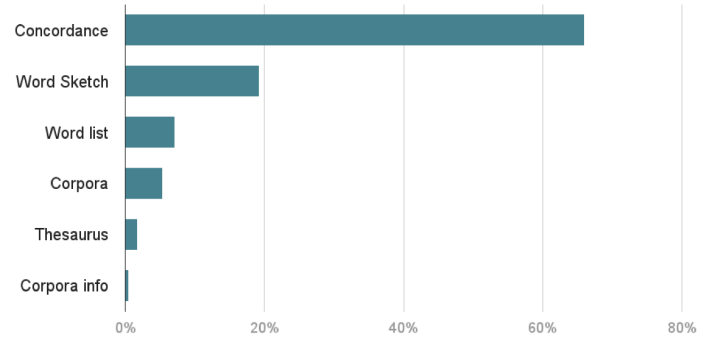


Figure 2.2: Top corpora used in SkE

Figure 2.3 shows that the most used feature is clearly the Concordance with 65 % of all page-views. The Word Sketch function is the second with almost 20 % of all page-views and the remaining main functions are used less than 10 % of all times.

Figure 2.3: Most requested features



SkE had about 6,500 unique visitors per month in the past two years but the number is rising in the next half year, so the trend of unique users and number of visits seems to be rising as can be seen in Figure 2.4. The number of pages shown to visitors was almost stable near 600,000 through the whole year 2014 as depicted in Figure 2.5, where also all hits are outlined – any files, including pages, that were demanded from the server.

This short, yet valuable analysis of official and beta online websites will be employed in the examination of SkE usage after the final release of visualizations and will help to determine whether the visualizations are utilized by users and to what extent.

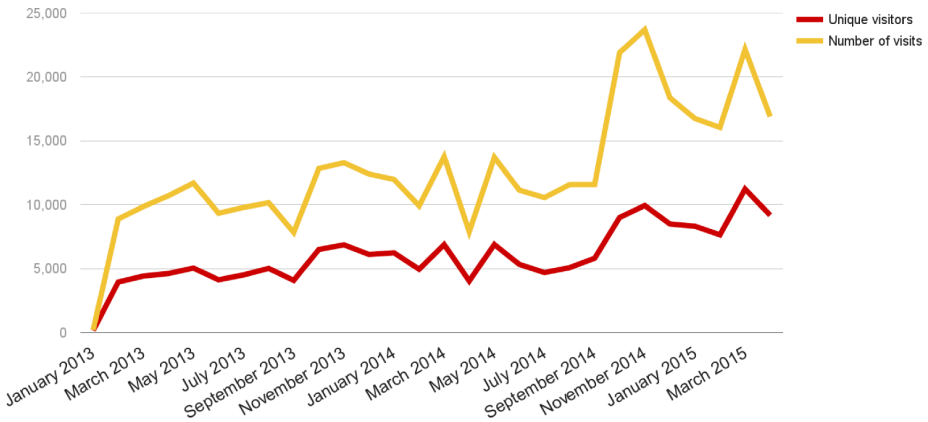


Figure 2.4: Number of visits and unique visitors

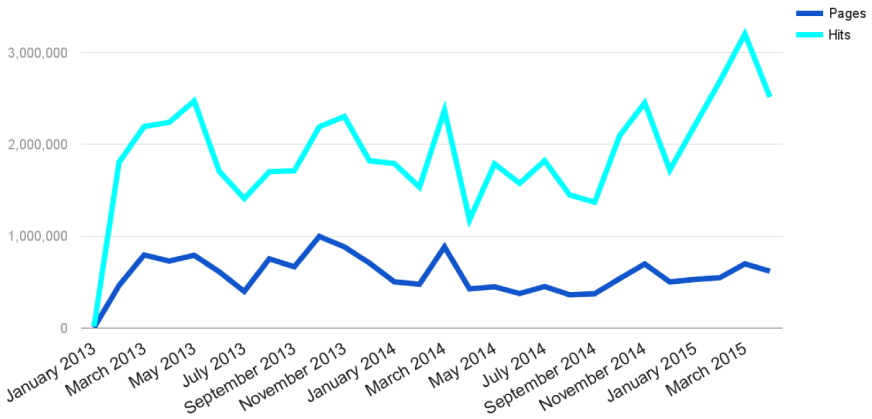


Figure 2.5: Number of hits and displayed pages

2.1.2 Features

Besides advanced corpora querying, SkE can also create a corpus from texts uploaded by users which can then be used in the system. However, this chapter focuses only on the core features for which visualizations will be designed within this diploma thesis. Detailed information and papers referencing SkE can be found in the documentation of the system[8].

[8] http://www.sketchengine.co.uk/documentation/

Concordance

Concordancer function shows chunks of a corpus where matches of the given query were found. The query can be as simple as one or more words, or it can be input in more complex expressions using Corpus Query Language (Christ and Schulze, 1994).

Query research **27,530** > GDEX **27,530** (245.40 per million)

HJ1	cities. The research will draw on a number of	research	methods. In simple terms the research will
HJ0	trends in applied economic and econometric	research	. The first is the development of various
HCT	indicate where they sourced their market	research	information. Over fourteen different providers
F9D	not a simple concept, and is difficult to	research	. So various market research interests have
G0R	from the rhetoric, what is the character of	research	in reality? Clearly, there are differences
FBM	the London Appeals Organizer for Action	Research	. He is 33 and lives in London with his
EF8	that experimental referential communication	research	is wasted effort. On the contrary, as Donaldson
J0U	distinction. The second area of current	research	in macroeconomics centres on a phenomenon
CL6	Kähler's insistence that faith and historical	research	are not the same thing is obviously valid
J2T	trading problems from time to time". Recent	research	has shown that the scrapie agent can live

Page 1 of 2,753 Go Next | Last

Figure 2.6: First ten matches of "research" query shown with its concordance

The results of such query are shown either as a whole sentence or in the form of keywords in context – known as KWIC. Figure 2.6 shows the default KWIC concordances of the query "research" in *British National Corpus*, where the matched words are highlighted in red and the identification code of the document on the left is blue. The function provides also numerous sorting and filtering options of the result, so the users can see exactly what they are looking for – for the selected example, first ten good dictionary examples (GDEX) (Kilgarriff et al., 2008) were chosen automatically by the system.

The system generates simple interactive visualization of frequency distribution. Figure 2.7 shows positions of all matches of "research" over the *British National Corpus*. This helps to easily identify documents containing the most or the least number of matches that can be then accessed by clicking on a specific bar in the chart. Normalizing the length of a corpus enables the comparison of the same query over different corpora.

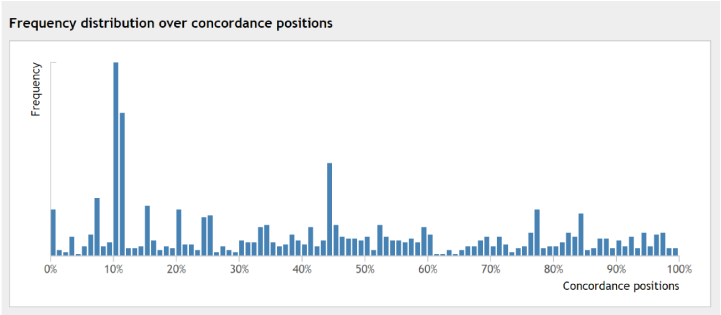


Figure 2.7: Distribution of "research" query in British National Corpus

Collocation candidates is a list of associated words in a given context of a query. The list is sorted according to a chosen association score, by default it is *logDice* (Rychlý, 2008), which represents different statistical importance. Even though, the system provides as many as eight different metrics – all of them are shown in Figure 2.8 – the analysis of user requests showed that users mostly use only *logDice* metric or the three metrics that are set as default (*logDice*, *T-score* and *MI*).

Collocation candidates

	Freq	T-score	MI	MI3	log likelihood	min. sensitivity	logDice	MI.log_f
P | N **solve**	1,203	34.657	10.331	30.796	15,874.394	0.022	9.418	73.285
P | N **solution**	832	28.730	7.980	27.381	7,667.462	0.015	8.772	53.667
P | N **with**	7,492	82.880	4.557	30.300	34,037.228	0.012	8.462	40.659
P | N **solving**	591	24.294	10.525	28.939	8,060.470	0.011	8.420	67.188
P | N **solved**	593	24.328	10.037	28.461	7,456.231	0.011	8.416	64.104
P | N **major**	823	28.286	6.157	25.526	5,442.299	0.015	8.416	41.339
P | N **cause**	711	26.428	6.817	25.764	5,358.445	0.013	8.411	44.773
P | N **serious**	692	26.085	6.894	25.763	5,290.443	0.012	8.393	45.092
P | N **arise**	584	24.099	8.500	26.880	5,834.389	0.010	8.342	54.159
P | N **no**	1,989	42.824	4.652	26.567	9,095.739	0.012	8.244	35.333

Figure 2.8: First ten collocation candidates for "problem"

Word list

Corpus: **British National Corpus**
Subcorpus: **spoken**

word	Freq
anything	6,257
next	6,093
new	6,062
long	6,059
always	6,020
saying	6,014
pounds	5,917
nice	5,887
eight	5,821
must	5,816

Figure 2.9: Few lines extracted from a word list generated from British National Corpus

Word List

Among other SkE features belongs Word List. It is a table containing list of words that are accompanied with their frequency – see Figure 2.9. It can be seen as a concise overview of a corpus which can be filtered with complex options, for example restricted it by a whitelist (words that must appear in the output) or a blacklist (words that should be omitted) or even a regular expression. (Kilgarriff, 2010)

Thesaurus

thesaurus – a type of dictionary in which words with similar meanings are arranged in groups (Cambridge University Press, 2015)

Thesaurus available in SkE represents so called distributional thesaurus – a list of words that occur in the same grammatical context (as defined in word sketches) as the given input word and hence they are likely to be semantically related synonyms, antonyms, hyponyms, hyperonyms or meronyms. The thesaurus function searches for words that share most contexts with the queried word from which a similarity score is calculated. (Rychlý and Kilgarriff, 2007) The exact value of this score is hard to interpret and its main use is to order the words that were selected. The output of the thesaurus, which can be observed in Figure 2.10 for a query "review", is a table containing a word accompanied by its score and frequency.

The thesaurus offers a sophisticated option to divide words into clusters according to words' distributional score. First, a list of words containing all pairs of words is created which is then used in the process of creating clusters. A word is put into already created cluster if it satisfies two conditions. First, the distributional similarity with the word and any other word from the given cluster must be greater than a threshold selected

by the user (set by default to 0.15 from a range [0,1]). Second, the similarity between the word and all other words within the given cluster must be greater than the similarity between the word and all words in other clusters. (Lexical Computing Ltd, 2010)

This function allows the user to differentiate many meanings of one word – as can be seen in Figure 2.11 which shows that the word "menu" can be used in completely distinct contexts.

As is illustrated in Figure 2.10 and Figure 2.11, the thesaurus displays a word cloud from the list. But here is font size mapped to the score value and not to frequency as would be expected. The colour is meaningful only when the clustering option is on – each cluster has its own colour, so similar meanings can be easily spotted.

For the thesaurus function, it is true that the bigger the corpus size, the better thesaurus will be created because similar words will be more clearly separated from other noise words. (Rychlý and Kilgarriff, 2007)

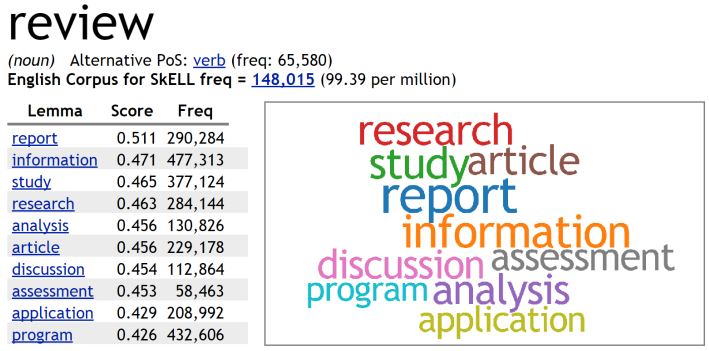


Figure 2.10: First ten words from thesaurus of noun "review"

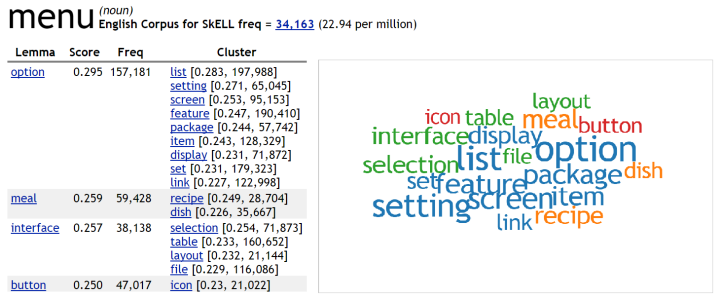


Figure 2.11: Clustered thesaurus of noun "menu"

Word Sketch

Word sketches were firstly created in 1999 for the purpose of compilation of *Macmillan English Dictionary for Advanced Learners* and were the trigger for establishing and further development of SkE (Kilgarriff et al., 2010).

drop *(verb)* Alternative PoS: noun (47,251)
English Corpus for SkELL freq = 137,036 (92.02 per million)

object of drop	54,449	4.50
bomb	2,088	9.18
anchor	566	8.01
charge	1,586	7.89
hint	370	7.38
ball	779	7.03

subject of drop	45,733	4.90
jaw	392	7.68
temperature	812	7.05
bomb	383	6.85
penny	108	6.00
price	646	5.83

modifiers of drop	22,705	0.70
dramatically	410	8.34
sharply	338	8.25
accidentally	253	7.79
precipitously	140	7.63
down	700	7.54

Figure 2.12: Three selected relations from word sketches of "drop"

The word sketch is an automatically generated list of collocates that the queried word is associated with in particular grammatical relation. For each grammatical relation that is available in the corpus a word sketch is calculated and summed in a table which includes a collocated word, its frequency and a score in every row. (Kilgarriff et al., 2004) Specific contexts of a collocated and queried word can be retrieved by clicking on a frequency number near the selected word. From Figure 2.12 we can find out that queried word "drop" creates a phrase with "jaw" and the most frequent collocates are "drop bomb", "drop charge" and "temperature drop". Alternative PoS (part of speech) is presented in the heading with its frequency, so the user can easily access other grammar forms of the same word.

This tool also provides complex filtering options including clustering that allows the users to observe and analyze a specific behavior of word meanings. Clusters are created in the same

way as in the thesaurus function but here the clusters are calculated for each relation separately, as shown in Figure 2.13.

Each table is sorted according to the *logDice* association score which represents a statistical measure of how relevant collocation candidates are. The *logDice* score is a modified *Dice* metric and is independent of the corpus size. Therefore queries from different corpora can be compared and the score is in a sensible range. An overview of other association score metrics and the detailed description of *logDice* are presented in (Rychlý, 2008).

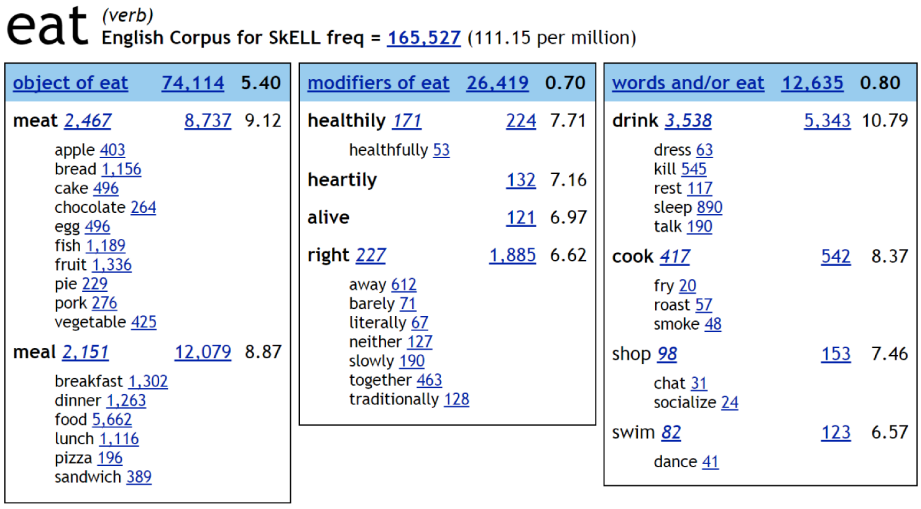


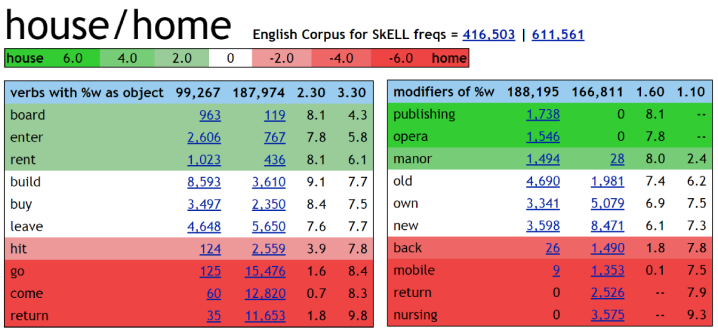
Figure 2.13: Three selected clustered relations from word sketches of "eat"

Word Sketch Difference

Word Sketch Difference, or shortly Sketch Diff, is a sophisticated yet easy way to compare two words. Similarly as in word sketches, each grammar relation is shown in the output as a table but each word has always two pairs of scores and frequencies.

A score difference is calculated for each word in a row simply as the score for the first query minus the score for the second query. Rows in the table are then sorted according to the score differences which defines whether the words in rows can be found with each of the queried words or with only one of them.

To easily distinguish the score difference, a colour range from green to red is assigned as can be seen in Figure 2.14. We can see that verbs such as "build" or "leave" can be used with both of the words "house" and "home" but a "publishing home" or "nursing house" are not found anywhere in the corpus, therefore using these phrases would not be appropriate. Detailed context can be also looked up with concordance after clicking on a frequency number as depicted in Figure 2.15.



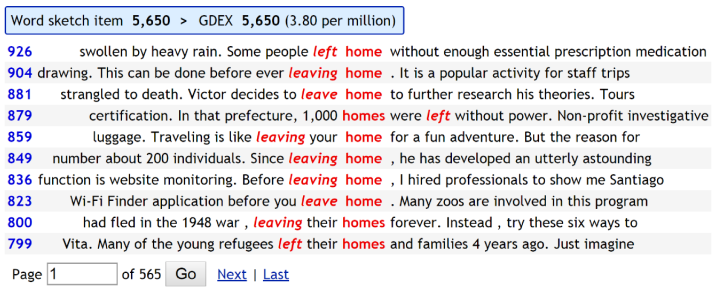
house/home English Corpus for SkELL freqs = 416,503 | 611,561

house 6.0 4.0 2.0 0 -2.0 -4.0 -6.0 home

verbs with %w as object	99,267	187,974	2.30	3.30
board	963	119	8.1	4.3
enter	2,606	767	7.8	5.8
rent	1,023	436	8.1	6.1
build	8,593	3,610	9.1	7.7
buy	3,497	2,350	8.4	7.5
leave	4,648	5,650	7.6	7.7
hit	124	2,559	3.9	7.8
go	125	15,476	1.6	8.4
come	60	12,820	0.7	8.3
return	35	11,653	1.8	9.8

modifiers of %w	188,195	166,811	1.60	1.10
publishing	1,738	0	8.1	--
opera	1,546	0	7.8	--
manor	1,494	28	8.0	2.4
old	4,690	1,981	7.4	6.2
own	3,341	5,079	6.9	7.5
new	3,598	8,471	6.1	7.3
back	26	1,490	1.8	7.8
mobile	9	1,353	0.1	7.5
return	0	2,526	--	7.9
nursing	0	3,575	--	9.3

Figure 2.14: Two selected relations from Sketch Diff of "house" and "home"



Word sketch item **5,650** > GDEX **5,650** (3.80 per million)

926 swollen by heavy rain. Some people *left* home without enough essential prescription medication
904 drawing. This can be done before ever *leaving* home . It is a popular activity for staff trips
881 strangled to death. Victor decides to *leave* home to further research his theories. Tours
879 certification. In that prefecture, 1,000 homes were *left* without power. Non-profit investigative
859 luggage. Traveling is like *leaving* your home for a fun adventure. But the reason for
849 number about 200 individuals. Since *leaving* home , he has developed an utterly astounding
836 function is website monitoring. Before *leaving* home , I hired professionals to show me Santiago
823 Wi-Fi Finder application before you *leave* home . Many zoos are involved in this program
800 had fled in the 1948 war , *leaving* their homes forever. Instead , try these six ways to
799 Vita. Many of the young refugees *left* their homes and families 4 years ago. Just imagine

Page 1 of 565 Go Next | Last

Figure 2.15: Detailed context of "leave home"

Corpus Info

Corpus info is one page summary of corpus metadata, for example the lexicon size or tags as shown in Figure 2.16. However, the corpus contains much more and detailed metadata that could be displayed. But because each corpus is unique and only some corpus families have common attributes, it is nontrivial to automatically summarize some features, especially themes.

Corpus *British National Corpus* - statistics and info

Counts	
Tokens	112,181,015
Words	96,048,950
Sentences	6,052,184
Paragraphs	1,514,906
Documents	4,054

General info	
Language	English
Encoding	UTF-8
Compiled	04/16/2015 14:49:50
Tagset doc	Description
Infolink	More info

Lexicon sizes	
word	773,598
ambtag	91
lempos	721,048
tag	61
lemma	674,529
lc	654,541
lemma_lc	620,586

Tags legend (tagset)	
adjective	AJ.
adverb	AV.
conjunction	CJ.
determiner	AT0
noun	NN.
noun singular	NN1
noun plural	NN2
preposition	PR.
pronoun	DPS|PN.
verb	V.*

Lempos suffixes	
adjective	-j
adverb	-a
conjunction	-c
noun	-n
preposition	-p
pronoun	-d
verb	-v

Figure 2.16: Statistics and summarized information about British National Corpus

2.1.3 User's description

Even though the SkE functions were firstly created and later developed for the purposes of experts in the field of lexicography, SkE can be used by almost anyone who is curious about language in general. Nonetheless, the main users of SkE can be divided into 6 groups:

1. Lexicographers are the main users of SkE as almost every main dictionary publishing company (namely Cambridge University Press, Harper Collins, Macmillan and Oxford University Press) use this system very frequently and so do many national language institutes. (Kilgarriff et al., 2014)
2. Other group of typical users contains universities where SkE is a part of an extensive research in many interdisciplinary fields, such as natural language processing or computational linguistics, but also in departments studying language as their main research object. (Kilgarriff et al., 2014)
3. Exploration of written and spoken language has an invaluable insight when it comes to second language learning. Recently, developers created more user friendly interface to new or not so advanced users called SkELL - Sketch Engine for Language Learning [9]. Students and teachers can explore the language, with the help of a specially prepared corpus named after the interface, without detailed knowledge of data nor the computation behind it (Baisa and Suchomel, 2014) as can be seen in Figure 2.17 showing the concordances of "learn". There are even novel approaches in teaching of languages with the application of the corpora tools. (Thomas, 2015)
4. Translators – a small yet very experienced group of users – look for special phraseology of a given domain which can be found in special corpora.

[9] https://skell.sketchengine.co.uk

5. Terminologists use SkE for observing certain terms in order to create consistent terms or validate the terminology and its usage. (Kilgarriff, 2013)
6. Several language technology companies use parts of SkE features as succour in their proprietary software. (Kilgarriff, 2013)

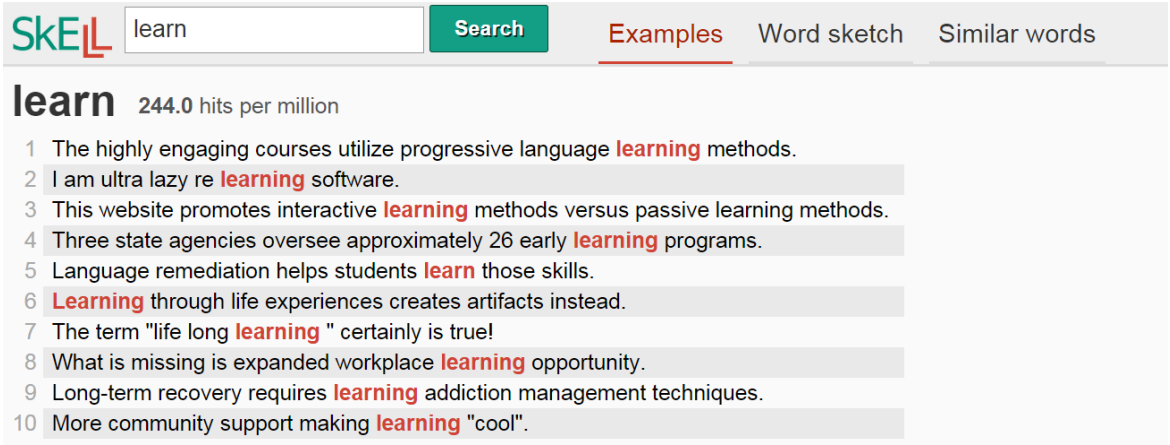


Figure 2.17: SkELL interface aimed at teachers and students

2.1.4 User's goals

Products designed and built to achieve business goals alone will eventually fail; personal goals of users need to be addressed. (Cooper et al., 2007)

As can be already seen in the statistics showed in 2.1.1, the main goal of all users is to see the actual examples of searched word or words in the selected corpus. And because users wanted to look for much more specific examples, the Corpus Query Language (CQL) was integrated into SkE to allow users to create complex requirements for the output.

Moreover, users are usually searching for phrases, idioms, and collocations of specific words. They also want to be able to compare words and their usage not only in one corpora but also in different documents of the corpora. The statistics are also helpful as the user wants to know whether the given result is common or should not be dealt with.

Overall, users want fast and reliable system that is always capable of giving relevant answers to their queries, so they can easily and conveniently evaluate the output of the system without wasting much time.

The goals are not mentioned or split within the features because the users can achieve the mentioned goals by using various features or can even combine them. The goals should not be interchanged with tasks, as tasks are activities that the user is taking in order to achieve an output – a goal (Cooper et al., 2007).

2.2 State of the art in corpora visualization

Research in corpora data analysis and visualization is nowadays thriving as need for quick perception of information and evaluation has never been more required as mentioned in 1. Many books (Steele and Iliinsky, 2010), (Meirelles, 2013), (Ferster, 2012), (Börner and Polley, 2014) are dedicated to visualizations from the general perspective but there are not specialized books or recent summary articles discussing visualization tools primarily designed for corpora – the reason is broader as Isabel Meirelles states in her recent (2013) book – *Methods and tools for the visualization of textual data are scarce. Examination of early books on visualization of information, including those by Willard Brinton, Jacques Bertin, and even Edward Tufte, reveal the lacuna. To my knowledge, the first book to dedicate a chapter on document visualization is Using Vision to Think by Card and colleagues in 1999. (Meirelles, 2013)*

Visualization tools for corpora can be divided into many possible categories – from employed graphic attributes to data structure. But this chapter presents them sorted by features that can be perceived from the generated graphics. However, it is not meant in all means to be an exhaustive summary but only as an overview of capabilities and features of visualization methods that are now available or usable for corpora.

2.2.1 Frequency summary of words

Simple summarization of word frequencies can be visualized with tools generally known as tag or word clouds in which font size depends on the frequency from the input text. The colour of words can be either derived also from their frequency, as in Figure 2.18, or the colour can be random, as depicted in Figure 2.19 which was generated by Wordle – one of the most popular word cloud tools.

Wordle firstly calculates the frequency of each word and according to its value a weight determining the font size is assigned. The angle of words can be adapted to user's desire – from strictly horizontal or vertical to fully random – and positions are generated using a randomized greedy algorithm that starts in the middle of the canvas and continues the positioning process with a spiral movement. If a given word collides with others, it gets another word from a stack and tries to put it in the selected position. Then it moves along the spiral and the placement strategy continues the same way. (Viegas et al., 2009)

Tag clouds use, in general, only the frequency as a meaningful attribute and as can be seen in given Figures 2.18 and 2.19, the spatial positions depend on the level of desired readability. Therefore, there are many additional graphical attributes utilized when the word clouds are adapted within complex systems.

Tag cloud visualization is already available in SkE feature Thesaurus, which was introduced in 2.1.2. But the existing visualization lacks mapping of score to any attribute and colours are not always meaningful. Therefore, the whole potential of data is not fully exploited.

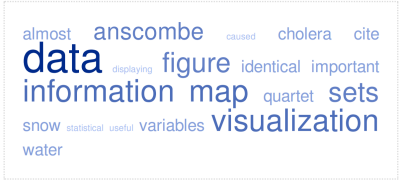


Figure 2.18: agCrowd with 20 most frequent words from first chapter

Figure 2.19: Wordle created from first chapter

2.2.2 Words relations

Relationships in text can be visualized in a network where vertices are represented by words and edges by a particular relation between vertices. This type of visualization is typical for WordNets[10] – *[manually created lexical databases] interlinking words and groups of words by means of lexical and conceptual relations represented by labeled arcs (Fellbaum, 2005).*

[10] http://globalwordnet.org/wordnets-in-the-world/

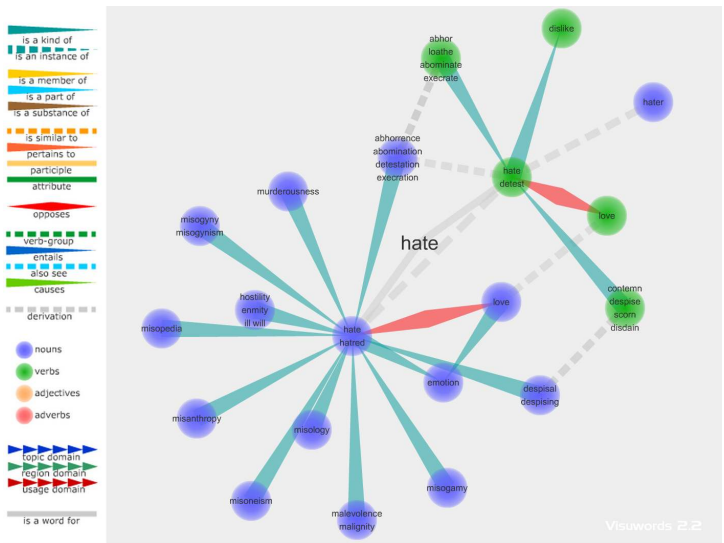


Figure 2.20: Visuwords online graphical dictionary showing net for a word "hate"

An open source WordNet created at Princeton University is used in Visuwords (Dunn, 2007) which can be seen in Figure 2.20 in which all types of relationships are shown in the left panel. The tool is available online at http://www.visuwords.com/.

Unstructured texts – without any metadata or markup – which are the usual textual information people come across, can be explored using TextArc (Paley, 2002) displayed in Figure 2.21. This system, unlike the others, is not generating only an overview of a given text but it respects the order of words. The entire text, except stop words, is displayed with small font on an eclipse line near to the border of a screen and within the eclipse a tag cloud is generated. Relationships are shown on demand – when a word from the tag cloud is selected, its distribution can be easily spotted on the eclipse and also detailed contexts are shown directly in the text window.

An approach to automatically reveal some relations from unstructured text was used in PhraseNet, which can either extract syntactic relations with the use of Stanford Parser[11] or with pattern matching. (van Ham et al., 2009) First method can take up to days of running, depending on the length of a text and its language. However, most of the corpora management systems provide already pre-processed sources, so an interactive exploration would be possible even without additional use of regular expressions which do not always analyze the text to its full extent.

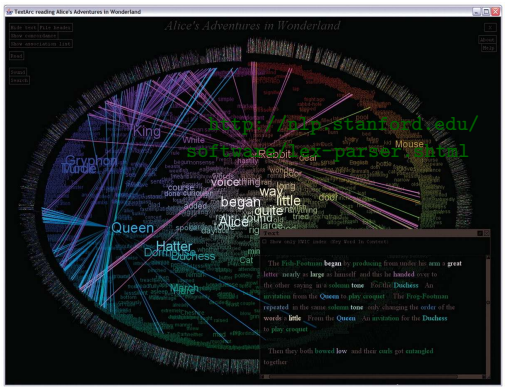


Figure 2.21: Alice's Adventures in Wonderland visualized with TextArc (Paley, 2002)

Figure 2.22 shows the final output from PhraseNet which was compressed and filtered after the patterns extraction. We can easily observe words which are mostly used in a relation "X of Y" as the frequency is mapped to font size similarly as in the previous visualization methods. Colours distinguish whether the word is object or subject of a given relation – the darker the blue the more outer relations it has and vise versa.

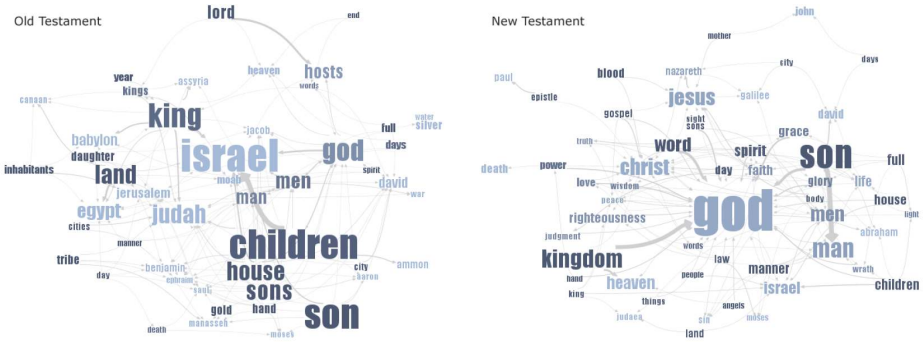


Figure 2.22: Phrase net displaying patterns "X of Y" in Old (left) and New Testament (right) (van Ham et al., 2009)

2.2.3 Concordance

Graphic representation of search results from text is a challenge – the reason discloses Marti A. Hearst: *[...] information visualization has not yet proven itself for search interfaces. It may be that the best uses have not yet been discovered, or it may be the case that the nominal nature of textual information renders visualization problematic for this particular application. (Hearst, 2009)* Representing the matching words with graphical elements is not always the suitable way from obvious reasons. The user wants to see mainly just the words – but it is necessary to find the right combination of graphics and words as the results from corpora, especially the huge ones, can have thousand or millions hits. Numerous interesting tools and methods has been proposed in the last years and some of them are presented below.

Figure 2.23: All occurrences of "love the" in King James Bible (Wattenberg and Viegas, 2008)

WordTree shows the full results of sentences within either a suffix or prefix tree structure which can be further refined by clicking on selected words. Figure 2.23 shows an example of such generated tree which reveals words that are used most often and their contexts.

The inspiration for creating another tool, DoubleTree, was the previous work of Viegas and Wattenberg. However, the WordTree tool lacked the visualization of suffix and prefix tree at the same time to provide a true KWIC view of results. (Culy et al., 2014) DoubleTreeJS is essentially the same tool but reimplemented using the JavaScript library D3 (Bostock et al., 2011). In Figure 2.24 we can see concisely visualized paths of syntactic structures of a query "see" which was then specified with "the"

by clicking on it. The frequency of words (only in a given syntactic position) is mapped to their size and the displayed strings are coloured red when they share at least one context.

Because the authors were aiming to apply this tool to corpora, they created not only various online demos[12] but also an example usage in Sketch Engine which provides the data and also the model for the tool[13].

interHist presents another approach for the evaluation of search results and is rather a complex tool, even though it resembles very simple bar chart. It is aimed primarily at expert users in the field of linguistics (Lyding et al., 2014). It creates a multiple bar chart, where each bar represents a position of a token in a given query and has a number of segments which display part-of-speech and their distribution for given token position. The legend on the right side lists all found part-of-speech tags and their colour code. Complex filtering can be used for further exploration but it can result in doubling the number of the bars which can be confusing. Figure 2.25 shows an example created with the Italian corpus in which the filtering by clicking on a particular bar was performed.

[12] http://www.sfs.uni-tuebingen.de/~cculy/software/DoubleTreeJS/

[13] http://www.sfs.uni-tuebingen.de/~cculy/software/DoubleTreeJS/developer_guide.html#queryEngine

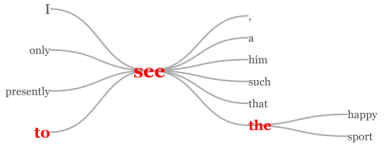


Figure 2.24: DoubleTreeJS showing results for "see the" from the book Robin Hood by Howard Pyle

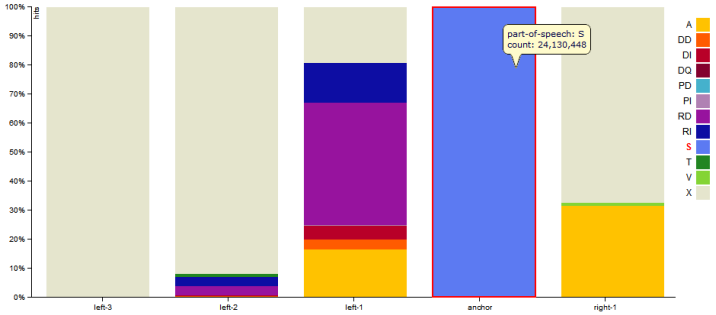


Figure 2.25: Results for noun phrases (Lyding et al., 2014)

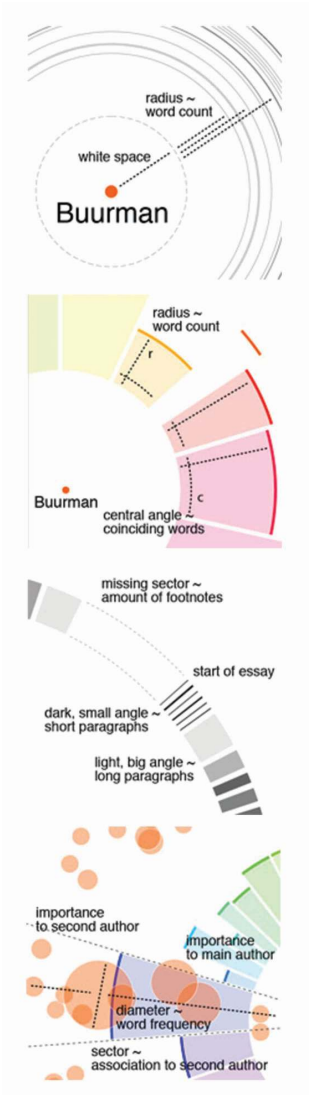


Figure 2.26: Mapping of different attributes in the Munterbund visualization (Rembold and Spth, 2001)

2.2.4 Differences and similarities of documents

Very desired functionality of any text processing system is an overview of similarities and differences between selected documents or texts. The amount of graphical elements that can be used is dependent not only on the number of samples that will be compared but also on the number of their attributes that should be represented in the output. Therefore, possible interactions with the system and the outputted graphics is almost necessary in order to give the user the full range of exploration capabilities.

A collection of visualizations was created for a book titled Total Interaction to facilitate the understanding of similarities in each of the nineteen essays. First, preprocessing of the essays is completed including removing stop words and other special characters. Then, similarities between the given essay and all the other essays are calculated which will determine the values of the graphical elements in the final output. The key graphical element is a word frequency mapped onto the circle's diameter and a distinct colour which was assigned to each author. Different lengths of the essays are mapped on the length of arcs. The number of coinciding words is mapped to its radius. As a supplementary information, there is a crosshatched ring which indicates the length of essay's paragraphs. (Rembold and Spth, 2001) Figure 2.26 illustrates in detail all the attributes that can be found in each visualization and one of the graphics created for the author Halter is shown in Figure 2.27.

In order to visualize the differences even in large text corpora that are rich in metadata, a combination of parallel coordinates and tag clouds was developed. As the source data a collection of over 600,000 court documents that covered 50 years was used in order to examine whether the court cases differ depending on their location. (Collins et al., 2009)

As can be observed in Figure 2.28, each column of words represents a subset of corpora and the size of words is mapped to their pre-assigned score. The same words in different columns are connected by lines, even when there is another column between them – this is an important difference in comparison with traditional parallel coordinates system. The width of lines can be different to emphasize the relative importance of connected words. Words in each column are sorted alphabetically, so searching through columns and also comparison of words distribution is easy.

The whole interface is fully interactive and the output can be amended to reveal detailed information, moreover getting more context of a selected word is possible through KWIC. (Collins et al., 2009).

The analysis of thematic alterations over time in texts is possible by using the visualization tool named ThemeRiver. It uses the metaphor of a river where each coloured flow' depicts an individual theme. There are several key notions that were conveyed – not only trends can be easily distinguished and compared but also the thematic strengths and their relationships can be indicated. Figure 2.29 illustrates a thematic flow of Fidel Castro's texts, in which we

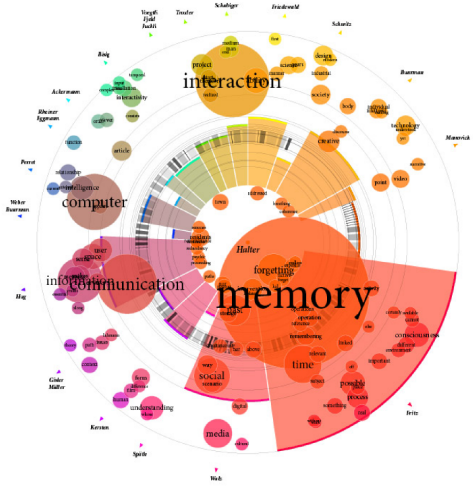


Figure 2.27: Comparison of essays by Halter and other authors (Rembold and Spth, 2001)

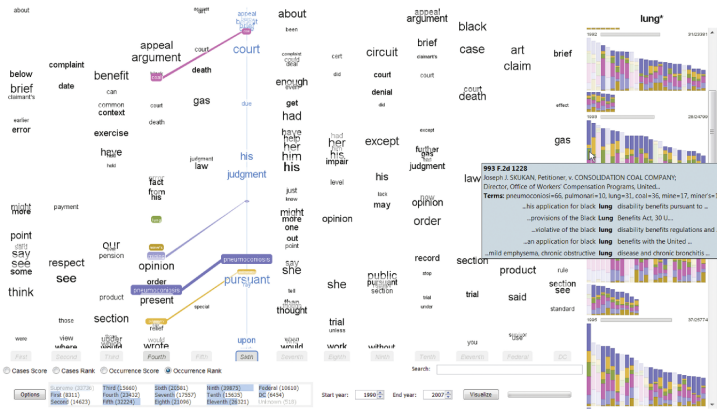


Figure 2.28: Overview of selected terms in parallel tag clouds (Collins et al., 2009)

can see that the subjects regarding weapons or reforms were almost constantly discussed but for example troops or oil topics emerged and soon disappeared. Above the flow there are added major events, so the user can also identify some causes or influences in the texts. (Havre et al., 2002)

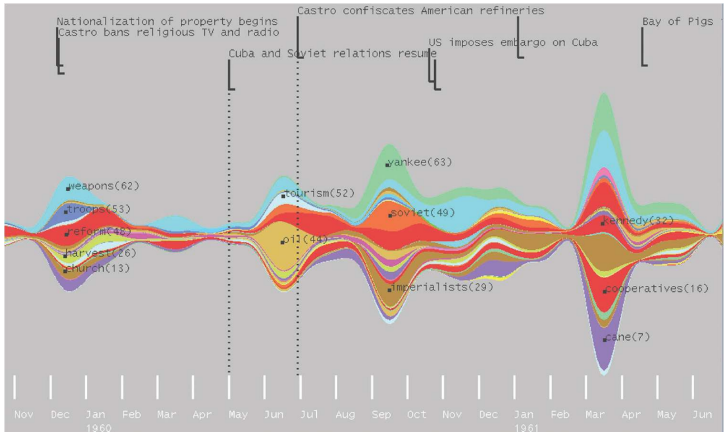


Figure 2.29: Collection of Fidel Castro's speeches, interviews and articles visualized with ThemeRiver (Havre et al., 2002)

2.2.5 *Repetitions in corpora*

Repetitions in text collections are in some cases easy to determine in visualizations, for example differences and similarities as in previous subsection 2.2.4 were parallel tag clouds. However, in other cases the user is looking only for the repetitions, so new methods are needed.

TileBars is a tool for presenting not only repetitions but also matches of a query, so it can be also included in 2.2.3. The method utilizes the structure of the document and visualizes the number of attributes in one simple output. The relative length of the document is presented with different lengths of "boxes" which include squares of the same size. Each square represents a section of the document and its colour is dependent on the frequency of the given section. Therefore, *TileBars simultaneously and compactly indicate relative document length, query term frequency, and query term distribution. (Hearst, 1995)*

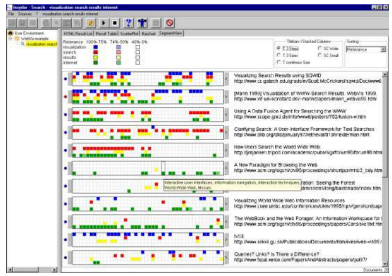


Figure 2.30: TileBars used in INSYDER (Reiterer et al., 2005)

This method was implemented as part of a system titled INSYDER: a content-based visual-information-seeking system for theWeb (Reiterer et al., 2005) which also added the colour to the tiles to separate different concepts, as shown in Figure 2.30.

Figure 2.31: FeatureLens showing a collection The Making of Americans (Ruecker et al., 2009)

FeatureLens incorporates four different methods that help the user to find repetitions in a given document. The main window, which can be also seen in Figure 2.31, offers frequent patterns in the left column, frequencies of the

selected words over the document on the top, the document overview displaying the original text with highlighted words and in the middle, the section overview visualizes phrases with coloured rectangles in the context. (Ruecker et al., 2009)

Repetitions in texts can be similarly visualized in dialR tool but instead of graphs or colour-coded lines it uses series of radars (Figure 2.32) where the user can set a query and watch how close the matches are when searching through documents.

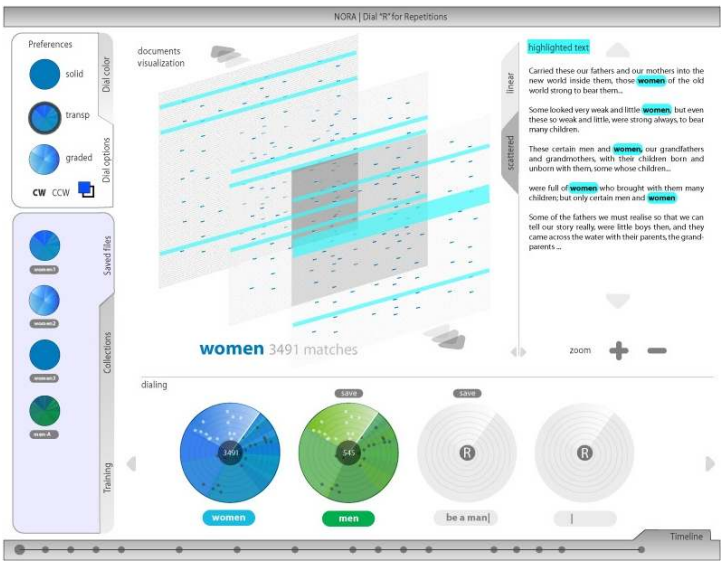


Figure 2.32: dialR with highlighted words (Ruecker et al., 2009)

2.2.6 *Topic modeling*

Automatic topic modeling – not only as visualization method but also in general – is crucial when documents do not supply enough metadata about their content and manual analysis would take too much time. Visualizations of topics facilitate choosing the right document in cases where the analysis heavily depends on the right selection of documents.

Textexture is build as a non-linear reading machine to get a glance at the main topics of a given text. In the created network, which can be seen in Figure 2.33, every word is a node and nodes are connected only if there is a relationship between them – nodes are not necessarily the most frequent words. The relationships between nodes are counted based on the paragraph and sentence structure; also Landscape reading model (Van Den Broek, 1995) is considered. Resulting network is then encoded into a graph, where the size of nodes is calculated according to their betweenness centrality – betweenness centrality is equal to the number of shortest paths from all vertices to all other vertices that pass through the given node. A node with a high value of betweenness has a large influence in the given network and node's colour is given according to a community the word belongs to. (Paranyushkin, 2011)

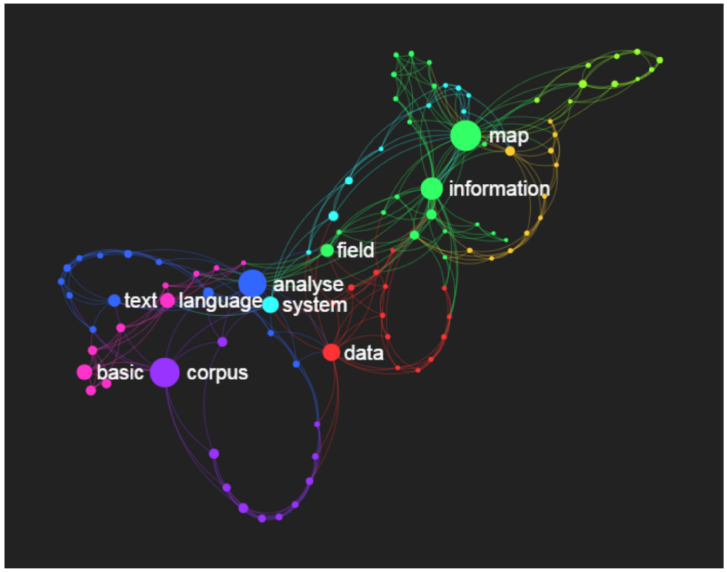


Figure 2.33: Network generated by textexture from first part of Introduction 1

The implementation uses Gephi[14] to encode the network

[14] https://gephi.github.io/

[15] http://sigmajs.org/

[16] http://textexture.com/

into a graph and the final visualization is created by sigma.js[15] library. The tool is available online[16].

Termite is a tool that uses tabular layout to visualize the topic models which helps to compare different latent topics. The authors also *[contributed] a novel saliency measure for selecting relevant terms and a seriation algorithm that both reveals clustering structure and promotes the legibility of related terms (Chuang et al., 2012)*, as can be seen in Figure 2.34.

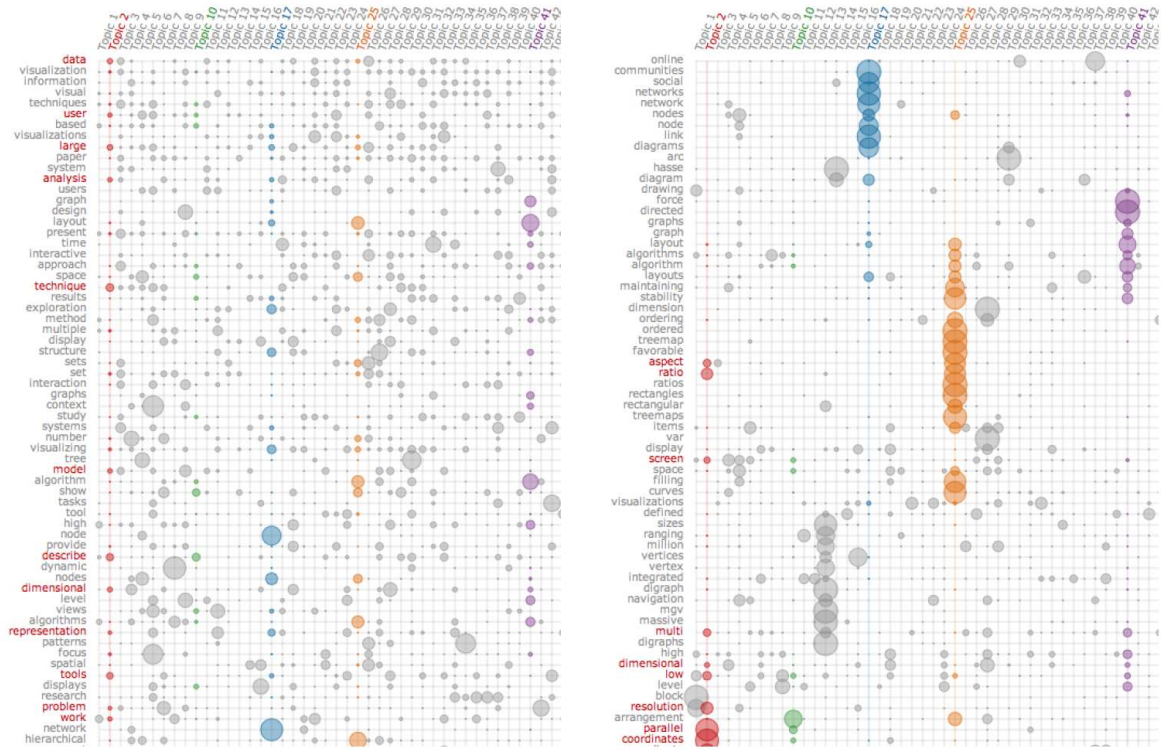


Figure 2.34: termite illustrating ordering by frequency (left) vs. seriation technique (right) (Chuang et al., 2012)

2.2.7 Statistical overview

Statistical overview of documents, and especially of corpora, is another primarily criteria when choosing the right text source for analysis. Even a simple visualization can help to better comprehend and easily interpret the statistics about the source.

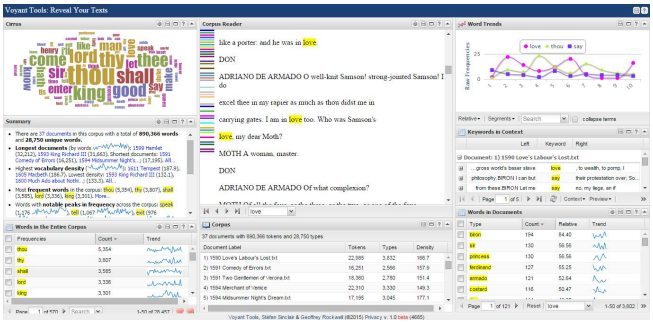


Figure 2.35: Overview of corpus of Shakespeare's plays in Voyant Tools

Voyant tools is a web-based interface[17] which was designed for analysis and review of documents. It combines the number of statistics tools, which can be seen in Figure 2.35:

- Cirrus (top of left panel) – word cloud with random colour,
- Summary (middle of left panel) – automatic statistics of corpus in a form of sentences,
- Words (bottom of left panel) – list of words with frequencies and trends,
- Corpus reader (top of middle panel) – overview of raw data,
- Corpus (bottom of middle panel) – list of documents,
- Word trends (top of right panel) – graph of frequencies over documents,
- KWIC (middle of right panel) – concordance overview with matched words,
- Words in documents (bottom of right panel) – list of words with selected attributes.

[17] http://voyant-tools.org/

3
Design

This chapter introduces a number of concepts for each of the main features introduced in section 2.1.2. The concepts were created to work as visualizations at their own but some of them were designed to work together within one system, so they use the same elements in order to remain consistent. The consistency within a system is critical especially when introducing new methods and features to users, because the overall consistency affects the speed of interpretation and comprehensibility. The more predictable and familiar the elements or functions of the system are, the more consistent the system is as a whole. (Johnson, 2010) All exact numbers would not be left out of the visualizations but would be available on demand, for example when the user hovers over an element. Various filtering options and altering of query attributes would be also at disposal, though this will be discussed in 5 as it is not directly related to the visualization methods of mapping the data to graphic elements.

Figure 3.1: Mosaic concept for corpora list

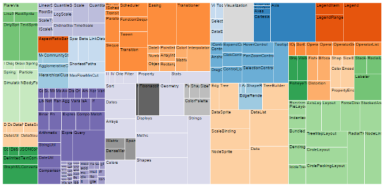


Figure 3.2: Treemap visualization example (Bostock, 2015a)

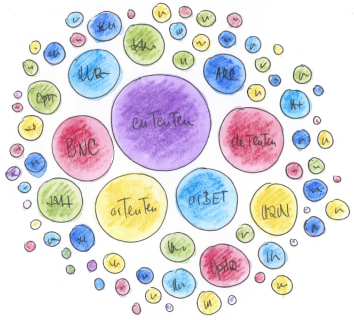


Figure 3.3: Circle concept for corpora list

3.1 Corpora concepts

Visualizations for corpora can be created either for a list of available corpora or for a family or subcategory of corpora which share the same attributes.

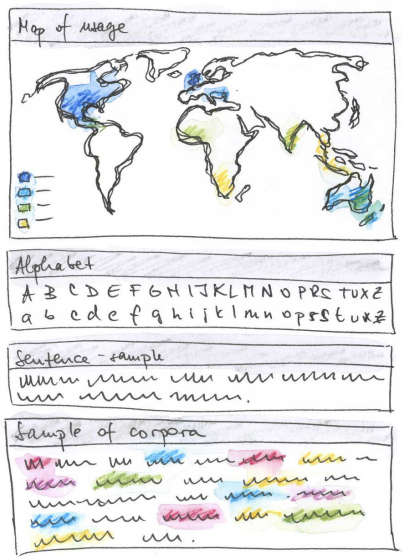
Interactive list of corpora can be made as a mosaic of all corpora which would be filtered automatically depending on the user's search input. The mosaic pictured in Figure 3.1 can be created from rectangles resembling famous Mondrian paintings. In contrast with the paintings, this concept does not fully cover the whole drawing area intentionally in order to not create a similar output as a treemap, which is pictured in Figure 3.2. The width of the rectangles would be mapped onto the number of tokens and the height onto the number of words, so the square-like elements will be easily recognizable as sources with similar number of tokens and words. Colour of each rectangle represents a given a language.

Another concept, shown in Figure 3.3, maps only one attribute of a corpora to the size of a circle – it can be either the number of tokens or words. It is more straightforward than the one using rectangles, especially when new users have no knowledge about the token meaning or what is the difference between a token and a word.

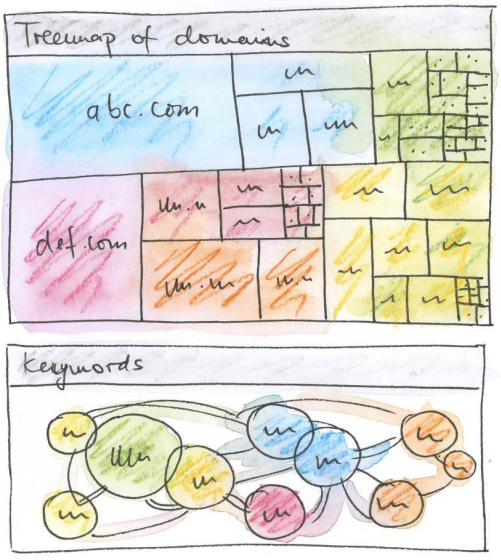
Figure 3.4 (a) shows the visualization combined with statistics that would have to be partially created manually – the map of language usage or characters forming an alphabet – but sample sentences or sentences with most frequent words could be automatically mined. This overview would be helpful for users who are interested in learning new languages or the ones that do not have many sources.

Corpora that was created by web crawling is rich in automatically generated metadata which could be exploited to create a treemap of domains. This feature would show not only

the variety of domains but could also help users to filter out domains they do not want to see, as illustrated in Figure 3.4 (b). Keywords stated in meta tags would create a visualization of thematic areas which would be colour coded for an easier manipulation.



(a) same language corpora



(b) web crawled corpora

Figure 3.4: Concepts for overview of

Visualizations for corpora families or a selected subcorpora can be created as a set of informative overviews that would encourage users to interaction and further exploration. Figure 3.5 shows attributes that could be automatically extracted from each of the already available corpora. For example, basic statistics of the source, thematic areas which would be colour coded and the same colours would be used for overview of sentence lengths or structures used in data.

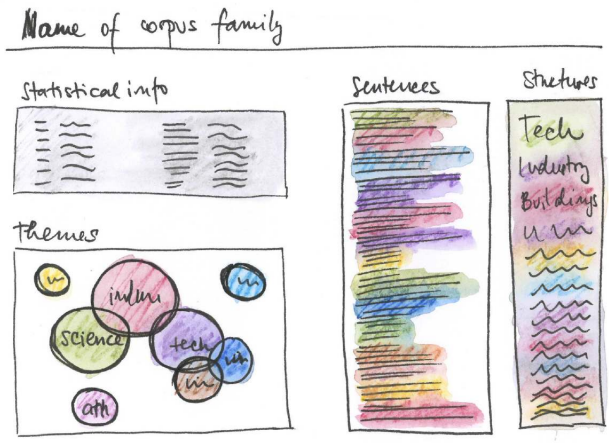


Figure 3.5: Corpus statistics with automatic created attributes

Concept that was chosen for the implementation in the scope of this thesis was the overview of corpora list depicted in Figure 3.3, as it was consistent with the concepts that follow later in this chapter. Moreover, it can be developed without any modifications or special preprocessing of data within the current system. In the future, a treemap and thematic areas will be implemented as they will be helpful not only for new users but also for the existing ones and could bring new perspective when exploring the meta information.

3.2 Concordance concepts

Users of SkE request the concordance to see words within sentences, so it is questionable whether they would appreciate at all a visualization that abstracts all the words. The types of tasks that would be suited for such visualization need to be firstly examined in detail in order to create a convenient and comprehensible tool. Therefore, the concepts for the concordance feature within this thesis will focus on displaying additional information that would be build upon the existing sentence structure.

First concept, which can be observed in Figure 3.6, shows that the original key words in context (KWIC) could be extended with two types of information – the word syntactic dependencies drawn above the words and the part-of-speech drawn below each word. The dependencies are drawn as oriented arrows, so it is clear which word is the head of a given relationship and parts-of-speech are colour coded within squares.

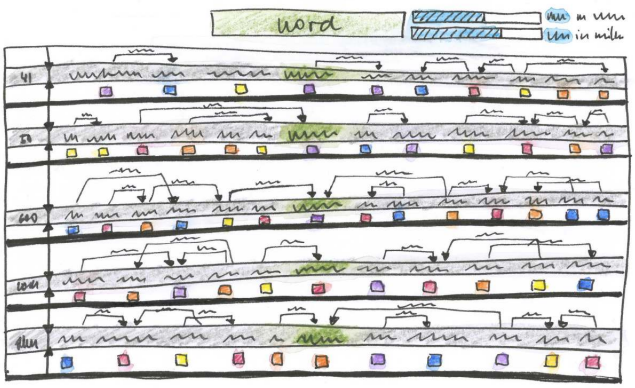


Figure 3.6: Concordance with added features

Figure 3.7 illustrates that the part-of-speech can be extracted from the results for easier pattern searching. But because looking through many rows of data is tedious, the patterns could be shown with a structure that is similar to a DoubleTree, as illustrated in Figure 3.8 – the frequencies of pairs could be displayed either with a bar chart next to each coloured square as in (a) or could be mapped to their sizes as in (b).

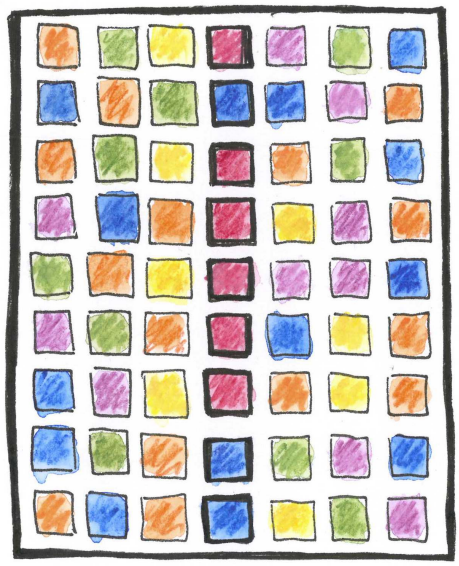
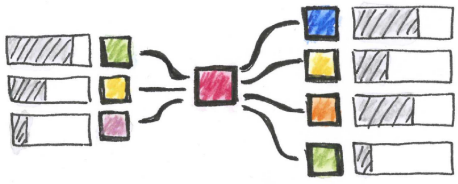


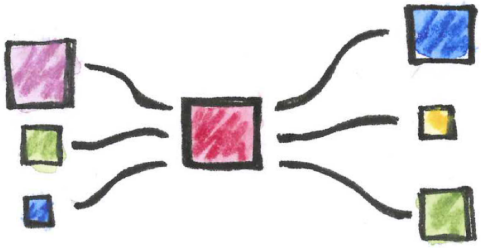
Figure 3.7: Part-of-speech shown without words

As illustrated in Figure 3.9, the exact representation of the relationships of part-of-speech could be even displayed as a network where the queried word is in the middle and the different categories that can be found following the queried word are forming levels around it. The main disadvantage of this visualization is that it displays only prefix or suffix network. Also, in cases when the words have many distinct neighbors, the network can be too complex to remain readable as is in (a).

For easier pattern searching and evaluation of the results, different tools could be put together within the same interface, so the user does not loose the track of given context and can interactively filter the displayed data. This concept combining different tools can be seen in Figure 3.10, specifically the results shown as KWIC as the main feature and additional information shown on the right side – the part-of-speech tree (top), DoubleTree (middle) and the frequency distribution (bottom).

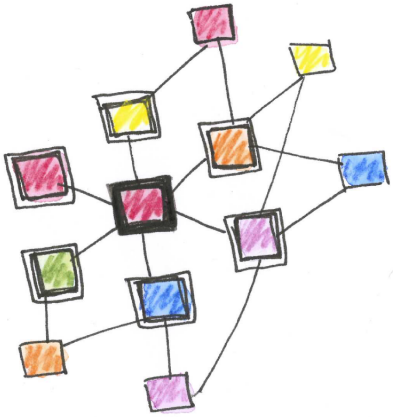


(a) displayed with horizontal bar chart

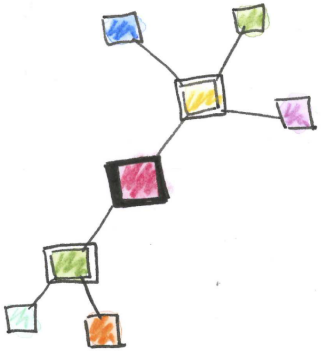


(b) displayed within the size of the square

Figure 3.8: Frequencies of part-of-speech pairs



(a) various types of relationships



(b) small number of relationships

Figure 3.9: Part-of-speech binding displayed as a network enabling to find

The developers of SkE were planning to implement the word sketch dependencies into the concordance, but because not only this particular concept but also the others introduced above require the nontrivial preprocessing of data or changes in the system in order to get the desired output, any of the presented concepts will not be implemented within the scope of this thesis but in the future.

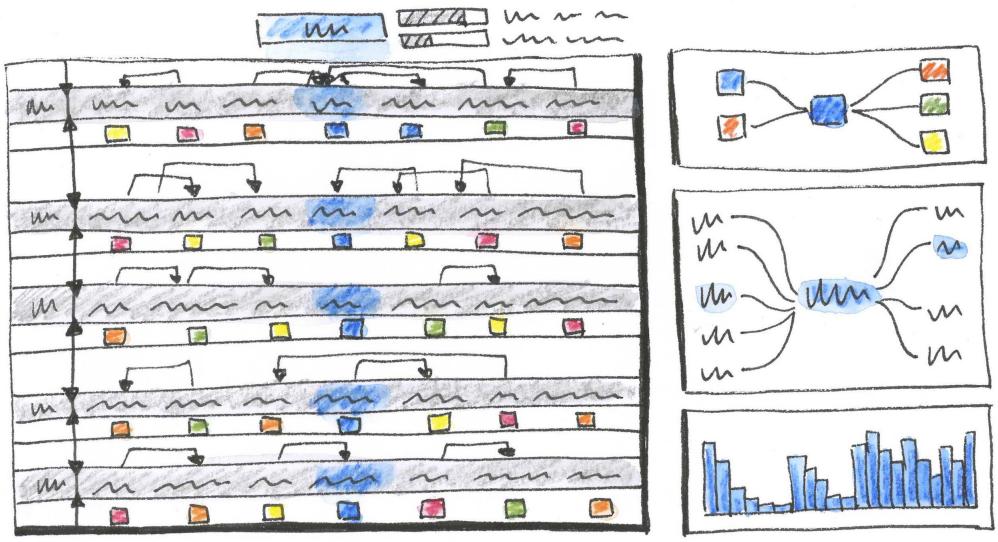
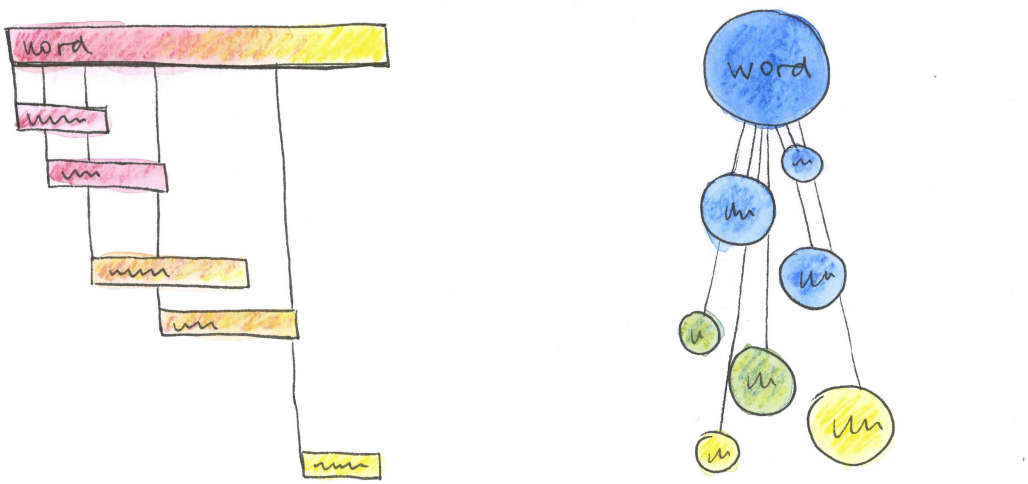


Figure 3.10: Different tools in one interface

3.3 Thesaurus concepts

The main goal of all thesaurus concepts is to represent the similarity as a closeness to the queried word. Moreover, an effective placement of words in the drawing area was also taken into consideration during the design process.

Figure 3.11 (a) shows a draft of the thesaurus which preserves the linear order of words and the overall appearance of a list that is already available in the system. The frequency is represented by the width of the rectangle and the score by the length of lines and also the colour. This representation is suitable only for a few words because the more words displayed, the longer the whole graphics would be.



(a) with the use of rectangles

(b) with the use of circles

Figure 3.11: Concept preserving list appearance

Another concept, which can be seen in Figure 3.11 (b), uses the same idea as above but replaces the rectangles with circles to achieve more straightforward interpretation of frequency. When there are any attributes mapped to the size of circles, they must be scaled to the circle area because any other method leads to misinterpretation as illustrated in Figure 3.12.

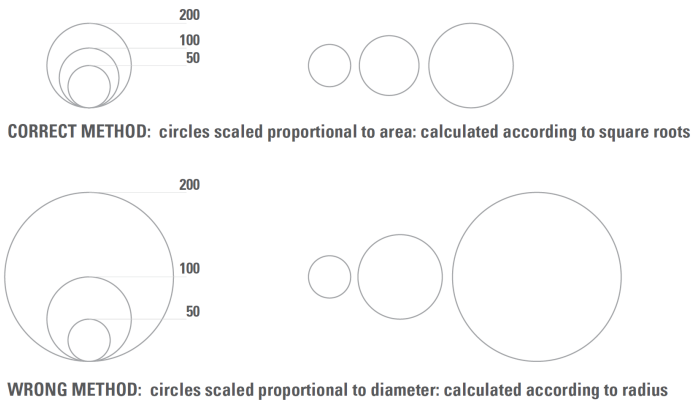


Figure 3.12: Scaling of circles (Meirelles, 2013)

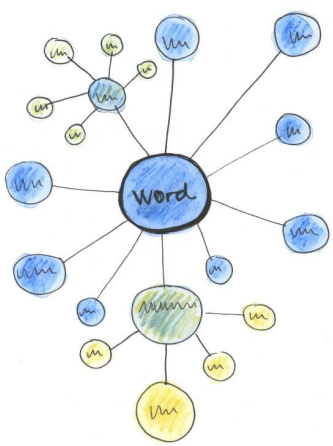
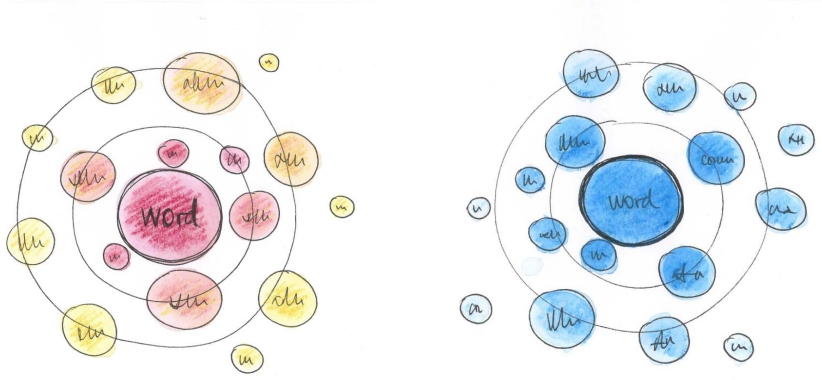


Figure 3.13: Circle concept for corpora list

Figure 3.13 stretches the previous concept into the whole area around the center, so the visualization covers the drawing area more continuously. The specific context of each word can be expanded, so the comparison of two similar meanings is possible within one frame.

The concept can be altered, so the output does not look like a manually created thesaurus and allows more loose representation of score. Figure 3.14 illustrates this method with one-colour scale (a) and two-colour scale (b). Colour scales can contain even more colours but it is necessary to test whether users can distinguish three or four-coloured ranges and also if they find them helpful.



(a) one colour scale using brightness

(b) two colour scale taking advantage of blending

Figure 3.14: Thesaurus as concentric circles

Figure 3.15 presents the combination of a circular bar chart and a list. This representation does not fully exploit the potential of the spatial distribution but on the other hand, it is very easy to read and interpret due to the fact that this kind of visualization is well-known and commonly used.

The concept shown in Figure 3.14 (b) was chosen for implementation as it makes the use of the drawing area effective and represents the score values intuitively. Specific colour scales will be chosen during the implementation but will be useful if users can select their own colour range.

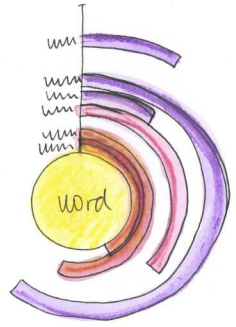


Figure 3.15: Thesaurus list combined with circular bar chart

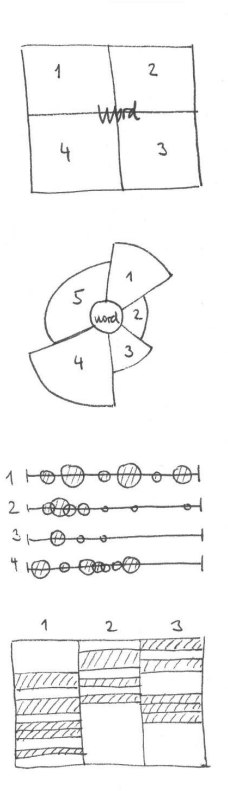


Figure 3.16: Different types of general concepts

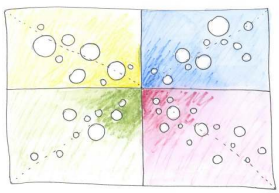


Figure 3.17: A set of four relations

3.4 Word Sketch concepts

There are different types of visual concepts that can be applied to the word sketch feature. A selected set is shown in Figure 3.16 where the top two concepts divide the space around the center and the bottom two concepts expand either to bottom or to one side of a window. As in the thesaurus, the score is represented as the closeness to either the center or one side of the visual output.

Figure 3.17 maps four relations into the same sized rectangles, so the comparison of different words is easy to perform. However, the user would be forced to choose always four relations, so this concept would not work in cases when the user wants to analyze either one or more than four relations.

The user should have a choice which particular relation or relations he or she wants to see, so the visualization should definitely support such requirements.

Because the word sketch is – strictly from the data point of view – just a multiple thesaurus table, the visualization can be build as a number of thesaurus visualizations that are combined together. Figure 3.18 (a) shows five relations with words represented within the frequency circles. Each relation is displayed as a rounded triangular segment where the overall score is represented by the radius and the overall frequency is mapped to the shape's area.

Each segment in the visualization shares the global score and the frequency scales, which means that the boundary values defining the scales are calculated from all words that are currently shown and not independently for each relation. This allows a quick comparison of different attributes at one time, especially the comparison of words from different relations.

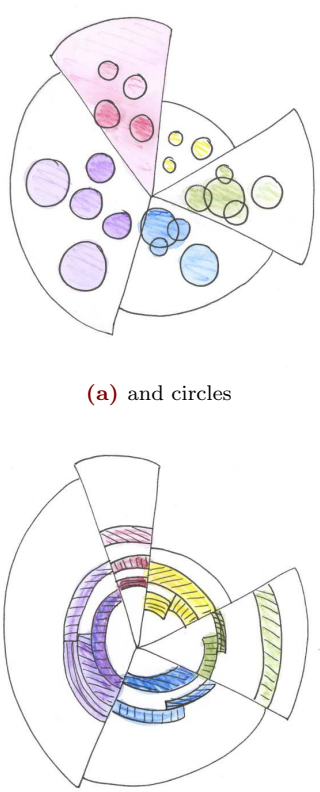
Apart from that, the user would be allowed to select specific relations he or she wants to see, so the visualization will always provide enough desired data.

Figure 3.18 (b) shows an alternative concept that also represents the relations by segments around the center. But instead of circles, it uses rounded rectangles for representing the word frequency.

The overlapping of circles was considered not only for word sketch but also for thesaurus. The possible output can be seen in Figure 3.18 (a) in green and blue segments, where the circles partially overlap. This can produce graphically appealing and interesting images, but because people tend to perceive elements that are near to each other as different groups (Meirelles, 2013), it would lead to misinterpreting the data because the overlaps would not be meaningful. Therefore, this possibility was not explored further in any of the presented concepts.

To keep consistent graphical representations throughout the whole system, the visualization presented in Figure 3.18 (a) was chosen for the implementation as it used the same mapping principles as thesaurus. Additionally, it was easier to interpret and also clear to use with various numbers of words.

Placing the words according to their similarity not only to the queried word but also to other words was discussed with the developers – it would be possible, though not guaranteed, to calculate such an output. But it would be too elaborate as the users are looking primarily at the queried word and if they want to distinguish between the groups of meanings, they use the clustering function in thesaurus or word sketch.



(a) and circles

(b) and rounded rectangles

Figure 3.18: Concept combining five relations

(a) divided into tiles

(b) unified as two-colour gradients

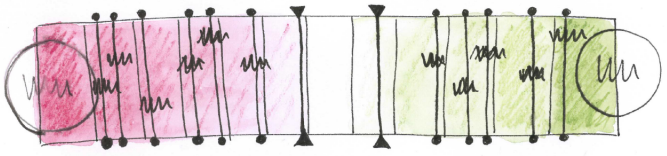
Figure 3.19: Layout of Word Sketch Difference

Word Sketch Difference concepts introduced below have a straightforward approach to visualize the results in a space between two queried words.

It is very similar to the output that the system currently offers but placing the words into two-dimensional space allows the user to immediately recognize whether the words tend to be used only with one or both of the queried words. If an outputted word is used with both words, it is usually positioned in the center of the visualization and the more it tends to be in contexts with only one of the queried words, the closer to the given word it will be.

Figure 3.19 shows two possible backgrounds of the visualization. It can be either divided into same sized rectangles for easier identification of the score value as in (a) or can be simple as two-colour gradient opposing each other as in (b).

Figure 3.20: Positioning of words in one relation



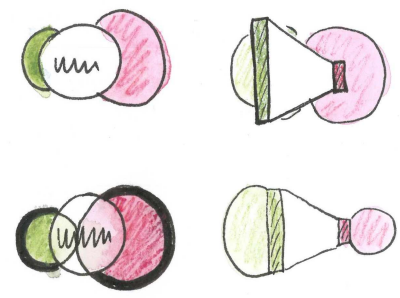
The concept of a layout shown in Figure 3.20 represents only one relation, therefore the final output of the system will be made of multiple rectangular visualizations.

The colour scales were taken from the current system but could be adapted according to user needs as red and green are

mostly perceived as a range of "bad" and "good" which is not applicable in this case. Moreover, people with sight difficulties will have problems to distinguish these two colours.

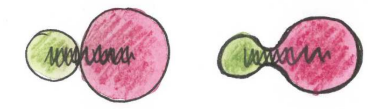
Figures 3.21 and 3.22 show various possibilities of elements representing words and their attributes. Frequencies are depicted as circles and scores are represented either with horizontal lines (3.22) or with vertical lines (b).

The proposed layout was approved as a good approach in visualizing this feature and the double circles illustrated in Figure 3.21 (c) were chosen as a representation for the word elements. The reason behind this decision was that the double circles represent well the fact that the frequencies are two different matters and they are not related nor sharing some properties as could be perceived by an element which is illustrated in Figure 3.21 (d). Displaying score values should be horizontal, not vertical, but should be turned off by default as it would make the visualization too complicated. The user will be still able to explore these elements on demand, though.



(a) overlapping circles

(b) combination of circles and lines



(c) two simple circles

(d) unified circles

Figure 3.21: Attributes of words displayed as

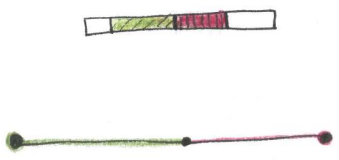


Figure 3.22: Possible display of score values

4
Technologies

An online application can be developed with various methods and tools that are usually combined to create the desired final product not only in a short period of time but also by reusing as much already available components as possible.

A brief overview of technologies that were used for the implementation of concepts introduced in 3 is stated in the subchapters below, including the reasons that led to choosing particular technologies for the online application. The emphasis was also placed on the complexity of further development or modifications when using the chosen technology.

All details regarding the implementation itself can be found in the next chapter 5.

4.1 Manatee/Bonito

Manatee is a corpus management system which can work with billion-size corpora and also offers a large number of lexical statistics. Bonito is a web interface for Manatee and SkE incorporates both of these technologies in order to deliver fast and reliable online service. (Rychlý, 2007)

The webpages of SkE are generated by Python-powered template system called Cheetah[1] and they also use various JavaScript libraries including jQuery and D3. Visualizations generated by D3 library are used for frequency distribution of concordance and thesaurus word cloud as described in 2.1.2.

[1] http://www.cheetahtemplate.org

4.2 HTML and CSS

Hyper Text Markup Language and Cascade Style Sheets are one of the oldest technologies in the web development. They are standards proposed by World Wide consortium and are constantly being evolved – HTML is currently in its fifth version and CSS level 3 has several modules published as formal recommendations.

HTML5 offers new features (elements, attributes, event handlers, and APIs) for easier web application development and more sophisticated form handling. (Jennifer, 2013)

Unlike previous versions of CSS, CSS3 isn't actually one single standard. As CSS has grown in complexity, the W3C has split CSS up into separate modules [...]. Since each module can develop independently of the others, there isn't any single standard called "CSS3." (McFarland, 2013)

4.3 JavaScript

JavaScript is the programming language of the Web. (Flanagan, 2011) It is standardized as ECMAScript and it is a dynamic and untyped programming language which enables developers to add behaviour to the content of their webpages. However, because it is a general-purpose language, it is not only used in web browsers but also in servers – one of the latest and most popular is Node which puts the emphasis on asynchronous operations and so enhancing the power of real-time applications. (Flanagan, 2011)

The JavaScript code can be as simple as one line added inline into the HTML page or it can be a complex application as for example Google's email client. Therefore, there is a wide range of libraries available that are supposed to ease the development of web applications especially by creating the reusable elements and hiding the variances of web browsers.

The most common client-side JavaScript library is jQuery. It is an open source tool which is *fast, small, and feature-rich [...]. It makes things like HTML document traversal and manipulation, event handling, animation, and Ajax much simpler with an easy-to-use API that works across a multitude of browsers. With a combination of versatility and extensibility, jQuery has changed the way that millions of people write JavaScript. (jQuery Foundation - jquery.org, 2015)*

JavaScript is also employed for drawing inside HTML5 Canvas. It is further explained in the following section 4.4.

4.4 SVG vs. CANVAS

SVG is a language for describing two-dimensional graphics in XML [XML10]. SVG allows for three types of graphic objects: vector graphic shapes (e.g., paths consisting of straight lines

and curves), images and text. Graphical objects can be grouped, styled, transformed and composited into previously rendered objects. The feature set includes nested transformations, clipping paths, alpha masks, filter effects and template objects. SVG drawings can be interactive and dynamic. (The World Wide Web Consortium (W3C), 2015b) Because the SVG elements are preserved in the Document Object Model (DOM), all elements are accessible. Moreover, event handlers can be assigned to them which allows to create rich application of SVG.

The canvas element provides scripts with a resolution-dependent bitmap canvas, which can be used for rendering graphs, game graphics, art, or other visual images on the fly. (The World Wide Web Consortium (W3C), 2015a) It is a powerful feature which can display not only 2D but also 3D graphics using WebGL. It can use SVG representation and commands.however, the final output will be still a bitmap. Therefore, when zooming into the image, it will be redrawn and can result in a blurry effect (Cecco, 2011).

When choosing between SVG and Canvas, there is no general answer to the question which of these tools is so called "better" as each of them serves its purpose and function.

The main advantage of SVG is that the data can be stored inside the Document Object Model as objects and their attributes. Moreover, the listeners can be bind to these elements which results in a code without unnecessary complexity. Canvas renders graphics from a pixel buffer. Therefore, the model is not aware of the shapes that are being displayed. (Microsoft Developer Network, 2015)

From the previous section, it is clear that when there are thousands or millions of graphical shapes that need to drawn, Canvas performs better than SVG because SVG needs to store

all the data inside the code. (Microsoft Developer Network, 2015)

In a manner of programming, it is questionable which approach is more suitable as every programmer has his or her own preferences. Both can be created using either pure JavaScript or additional libraries. For SVG, in addition to D3 there are also available snap.svg[2] or SVG.js[3]. The most popular libraries for Canvas include Processing.js[4] and fabric.js[5].

Recommendations about which technology should one pick can be summed up into several points, as stated in Figure 4.1.

[2] http://snapsvg.io
[3] http://svgjs.com
[4] http://processingjs.org
[5] https://github.com/kangax/fabric.js

SVG	Canvas
Vector image editing	*Interactive image editing: cropping, resizing, filters (think red eye removal, sepia, colorize, etc.)*
Highly interactive animated user interfaces	*Generating raster graphics: data visualizations, data plots, rendering fractals, function plots*
Data charts and plots	*Image analysis: read pixels to gather data for histograms, color usage, and anything else you can imagine*
Resolution-independent Web application user interfaces	*Rendering game graphics, such as sprites and backgrounds*

Figure 4.1: Criteria when SVG or Canvas is more suitable (Sucan, 2010)

Another important issue, the support of web browsers of the given technology, must be considered. Figure 4.2 demonstrates the ability to display SVG with basic functions and Figure 4.3 illustrates the similar for Canvas. Only versions of browsers which have at least usage of 0.55 % are shown in figures 4.2 and 4.3 (this excludes about 7 % of users).

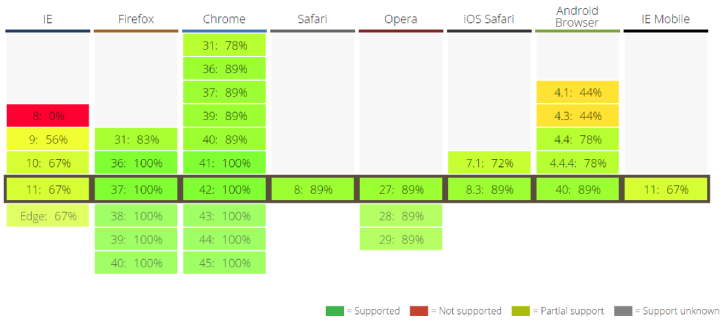


Figure 4.2: Browsers support of SVG and its main features (Deveria, 2015)

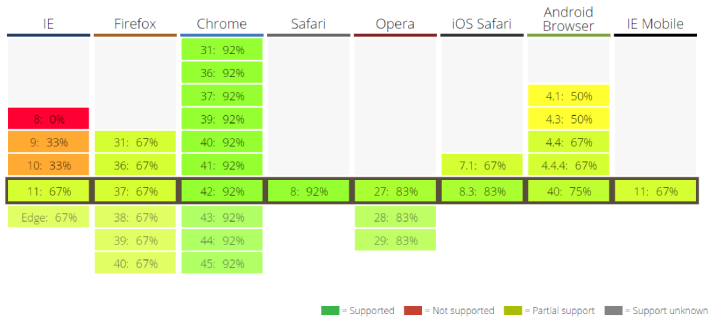


Figure 4.3: Browsers support of Canvas and its main features (Deveria, 2015)

As the final technology for implementation, SVG was chosen as it will be the most suitable form for the representation of SkE results – the visualizations will display mostly about 30 words, so there should not be issues with excessing the DOM hierarchy with too many elements. More importantly, the visualization interface will be highly interactive, therefore using SVG will allow for easy and quick manipulation of data.

4.5 Data-Driven Documents

Data-driven documents, or known as D3, is an open source JavaScript library that enables to bind the data of documents to elements of Document Object Model (DOM), which can be then animated by applying dynamic transforms when a data change occurs. The definitions of its main objectives – compatibility, debugging and performance – were created after developing and utilizing the Protovis (Bostock and Heer, 2009), an extensible toolkit for data visualization using SVG. D3 is complementing web standards such as SVG, HTML5, and CSS, therefore empowers the developers with utilizing the knowledge and resources. (Bostock et al., 2011)

D3 supports so-called "modern" browsers, which generally means everything except IE8 and below. D3 is tested against Firefox, Chrome, Safari, Opera, IE9+, Android and iOS. [...] D3 is not a compatibility layer, so if your browser doesn't support standards, you're out of luck. (Bostock, 2015a)

Figure 4.4 demonstrates the great potential of possible applications of D3 – it can be used for various graphical outputs ranging from simple graphs to rich interactive visualizations.

Because D3 serves as the general purpose visualization library, it has no build-in functions that can automatically create a particular graph type. Therefore, many tools were developed on top of it, for example NVD3[6] and C3[7]. Such tools require little or no programming knowledge, so they are perfectly suitable for quick and reusable data manipulation also for novices or users from different fields.

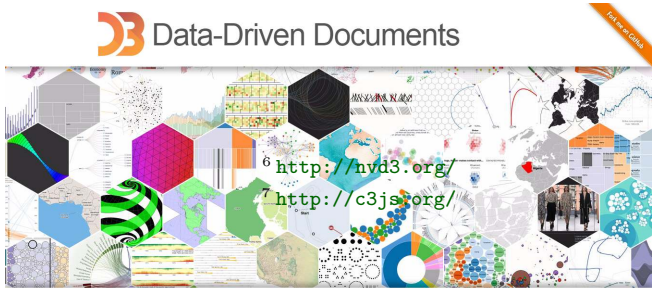


Figure 4.4: Demonstration of various possible applications of D3 (Bostock, 2015a)

4.5.1 Selections

A selection is an array of elements pulled from the current document. (Bostock, 2015b) Selections are atomic operands and there are two main functions – .select() and .selectAll(), where the former selects the first element that is found in the documents and the latter selects all elements that match the given selector. *D3 adopts the W3C Selectors API to identify document elements for selection; this mini-language consists of predicates that filter elements by tag ("tag"), class (".class"), unique identifier ("#id"), [...], and various other facets. (Bostock et al., 2011)*

An example of a selection is stated in Figure 4.5, where method chaining, that creates clear and short code, is also demonstrated.

```
// select all <circle> elements
// and set some attributes and styles
d3.selectAll("circle")
    .attr("cx", 20)
    .attr("cy", 12)
    .attr("r", 24)
    .style("fill", "red");
```

Figure 4.5: An example of selection and further method chaining

4.5.2 Data

The data operator binds input data to selected nodes. D3 uses format agnostic processing [13]: data is specified as an array of arbitrary values, such as numbers, strings or objects. Once data is bound to elements, it is passed to functional operators as the first argument (by convention, d), along with the numeric index (i). (Bostock et al., 2011)

When creating a simple static visualization, the code for creating the basic elements can be similar as in Figure 4.6. There is only a basic selection, followed by the append function which creates a new element with given attributes.

```
// simple static visualization
svg.selectAll("rect")
    .data(dataset)
    .enter()
    .append("rect")
    .attr({
        x: function(d, i) { return i*(w/dataset.length); },
        y: function(d) { return h-(d*4); },
        width: w/dataset.length-barPadding,
        height: function(d) { return d*4; },
        fill: function(d) { return "rgb(0, 0, " + (d*10) + ")"; }
});
```

Figure 4.6: Simple code for creating static visualization (Murray, 2013)

Although more complex instances are not much different from the previous example – as can be observed in the Figure 4.7 – there are few additional lines that simplify the automatic update of the whole visualization. Therefore, when data change occurs, the whole visualization is automatically updated with no requirements for manual update.

The data update process can be separated into three phases:

- enter – creates new elements according to the given specification,
- update – handles changes of existing elements,
- exit – contains references to elements that were not assigned to any data and removes them from the visualization.

```
// visualization with automatic updates

//Select...
var bars = svg.selectAll("rect")
    .data(dataset, key);        //Bind data with custom key function

//Enter...
bars.enter()
    .append("rect")
    // other attributes
    .attr("fill", function(d) { return "rgb(0, 0,"+(d.value*10)+")"; });

//Update...
bars.transition()
    .duration(500)
    // other attributes attributes
    .attr("height", function(d) { return yScale(d.value); });

//Exit...
bars.exit()
    .transition()
    .duration(500)
    .attr("x", -xScale.rangeBand())  // <-- Exit stage left
    .remove();
```

Figure 4.7: Code creating an automatically updating visualization (Murray, 2013), shortened)

Figure 4.8 illustrates these three phases and their handling of data change. New data, depicted in blue, are added to the existing data, depicted in orange. That means that elements E,F,G,H were altered, therefore their attributes will be updated. Besides, elements I,J,K,L will be removed and elements A,B,C,D will be added to the visualization.

In order to match DOM elements to data, the joining function needs to know which elements should be paired to which data. This is where a pairing key is applied – if a key is the same for a datum and an element, the datum can be then assigned to it. The pairing key is usually an index but when the order of data changes, it is no longer sufficient, therefore a function can be defined that will return an eligible key. (Murray, 2013) Figure 4.9 illustrates the possible usage of a key function when joining an array of word objects in which each words has a text attribute.

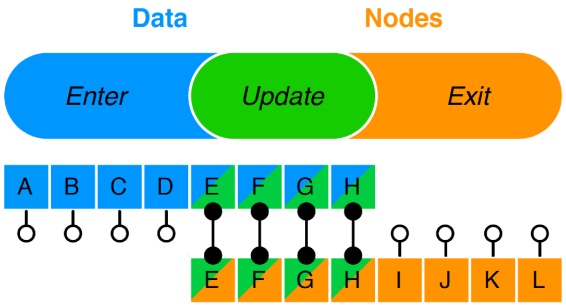


Figure 4.8: Data join logic (Bostock et al., 2011)

```
// e.g. words = [
//        {text:"an", val:3},
//        {text:"example", val:7}
//      ]
.data(words, function(d){
        return d.text;
    })
```

Figure 4.9: An example of a key function

4.5.3 Functions and interactivity

D3 offers wide range of functions that hide mathematical calculations and so ease the development of visualizations. The most often used are scales and axes. *Scales are functions that map from an input domain to an output range. (Bostock, 2013)* There are three types of scale functions – quantitative which have continuous input domain, ordinal scales which domain is discrete, and time scales. The visual representation of scales are axes which are also functions, therefore the numerous attributes can be used to alter them into the desired output. (Murray, 2013) Figure 4.10 illustrates the usage of both of mentioned functions.

```
//Create scale functions
var xScale = d3.scale.linear()
     .domain([0, d3.max(dataset, function(d) { return d[0]; })])
     .range([padding, w - padding * 2]);
//Define X axis
var xAxis = d3.svg.axis()
    .scale(xScale)
    .orient("bottom")
    .ticks(5);
```

Figure 4.10: An example of using scales and axes (Murray, 2013)

Interactions in D3 are made through the event listeners which are supported by the document object model. *[They are] callback functions that receive user input events targeted at specific elements. (Bostock et al., 2011)* Automatic animations are created with the transition() function. Because the data updates are handled by the function illustrated in Figures 4.7 and 4.8, the selection is able to determine what is the starting and ending point that should be used for the animation. Therefore, a default animation can be done just by calling the transition() function but additional alternations are possible.

4.6 Browsers accessing SkE

In order to make the visualizations available to as many users as possible, the browsers that are accessing SkE were analyzed and evaluated whether they support SVG.

The Figure 4.11 clearly shows that the majority of users will be able to display and interact with the visualizations. More specifically, about 85 % of users should have no issues and about 15 % of users will not be able to access the visualizations, which is caused either by the atypical browser type or using old versions of Internet Explorer. Its version 9, which is used by around 4 % of users, was decided not to be supported due to many issues with displaying particular elements and features with JavaScript. Less than 1 % of users accessed SkE through a mobile device. Therefore, the whole interface was not designed to be responsive but in the future there should not be problems with an alternation of the current implementation.

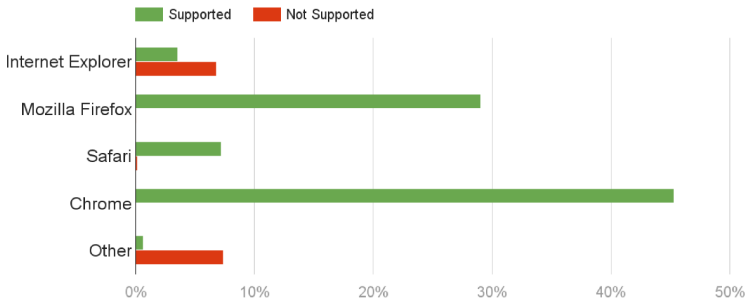


Figure 4.11: SkE users' browsers and their support of SVG

To notify the users with versions of browsers that do not support the needed features, a library called Modernizr[8] can be used – it detects whether the user's browser is capable of displaying the given figures and if not, an appropriate message is displayed to the user.

[8] http://modernizr.com/

5
Implementation

The following subchapters introduce the implementation details from the development of concepts presented in 3.

For the purpose of this thesis a local installation of Manatee/Bonito was prepared – the credentials for access are stated in 7.1.

The main aim of the implementation was to create scripts that were easy to deploy to any website, not only to the Sketch Engine interface. Therefore, the whole interface with the visualization and all controls is created dynamically at the page load. Additionally, only the visualization and the legend panels are re-generated when there is a new data request. Therefore the page is not being reloaded and the experience with the system is continuous and reactivity is instant in most cases (this is of course dependent not only on the speed of the server and scripts but also on the browser and machine that the user is using).

To embed any visualization into a webpage, two scripts are needed that were created within this thesis and a few additional scripts to provide the quality user experience.

5.1 Common workflow of visualization generation

Each visualization concept has a common workflow of the visualization and interface generation in order to propose the consistent approach and also to ease the future implementation changes. The workflow is as follows:

1. Initialization process:
   - 1.1. Variables that act as global are initialized (only this step is done for the corpora overview)
   - 1.2. A question panel with input boxes is created – when user inputs any data, the invoking process starts.
   - 1.3. If there is a full query given in the URL, the script parses it and creates the whole visualization by invoking the data.
2. Invoking data from server:
   - 2.1. Request for data in JSON is sent to the server with the passed variables (in the case of corpora overview a CSV file was parsed in the input).
     - 2.1.1. Because the request is an asynchronous method, the parsing of data and also call of visualization and controls generating method is done within the request function.
3. Visualization:
   - 3.1. A SVG panel is created in the DOM.
   - 3.2. Boundary variables are calculated from the data.
   - 3.3. Scales are set up according to the boundary values.
   - 3.4. Elements containing a text element and a circle element are tried to be positioned:
     - 3.4.1. If a text element or a circle element is overlapping with already placed elements, a new position is generated and the threshold value is increased.

3.4.2. If the threshold value is exceeded, all scales are scaled down proportionally so the relative relations remain the same.

3.4.3. If the scales are made too low as a cause of repeated scaling down, the elements are placed without any collision detection – it is mainly because the number of elements is too high thus impossible to position them without overlaps.

3.5. All elements are drawn onto the SVG panel (therefore also added into the DOM) according to the position generated in the previous step.

4. A control panel is generated and a listener is bind to each of the options displayed in the panel.
5. A legend is created according to the scales (this step is not done for the corpora overview).

5.1.1 Main common functions

All the visualization methods use circles and text elements within them. Because the visualization should be easily readable, collisions have to be calculated among other matters.

The script named vis_main.js contains common functions for:

- displaying error messages that was sended by the server,
- parsing and formatting of a given text, e.g., frequency values, in order to display them in a user-friendly format,
- generating points and angles for positioning of elements,
- calculating collisions of circles and text elements.

5.1.2 External scripts and templates

A number of external scripts were used in the implementation in order to bring the user a good experience when interacting with the system:

- jQuery[1] – required by most of the listed tools,
- jQuery UI[2] – used for control panels to provide complete functionality,
- Seedrandom[3] – added in the code in order to generate words always in the same positions but it can be omitted,
- Colorbox[4] – chosen for displaying the images with help in the current window,
- Colorpicker[5] – offered an easy implementation of user-friendly colour picker,
- SVG export[6] – added in order to allow the users to download the content of the SVG panel as a file, the source code is based on the demo of A. Gordon.

[1] http://jquery.com/

[2] http://jqueryui.com/

[3] http://davidbau.com/encode/seedrandom.js

[4] http://www.jacklmoore.com/colorbox/

[5] http://www.eyecon.ro/colorpicker/

[6] http://d3export.housegordon.org/

The templates of the webpages were altered to include two HTML tags (in case of corpora overview only one), to which the generated visualization with the interface will be attached – an element div for heading and another element div for the svg panel. At the end of the body, a hidden form for the SVG export was also added. The content of an element body for thesaurus function can be seen in Figure 5.1 and the content for other functions are almost identical to it.

```
#end def

#def main
  <!-- VISUALIZATION -->
  <div id="heading_thes"></div>
  <div id="svg_thes"></div>

  <!-- CODE IN FORM AND PERL SCRIPT WAS IMPORTED FROM  https://github.com/agordon/d3export_demo -->
  <form id="svgform" method="post" action="$files_path/misc/download.cgi" style="display:none">
  <input type="hidden" id="output_format" name="output_format" value="">
  <input type="hidden" id="data" name="data" value="">
  </form>
#end def
```

Figure 5.1: Body content for thesaurus function

5.2 Corpora implementation

The textual input for corpora visualization was created from each of the listed corpora at `corpora.fi.muni.cz` – all corpora, featured corpora, and parallel corpora. The visualization can be seen in Figure 5.2.

The corpora overview visualization generates for each corpus ten different positions that do not collide with already placed elements. Then, the most close one is picked as the final central point for the circle and text element.

Unique colour is assigned to each language, so the the different corpora of the same language can be distinguished. But because SkE currently offers 75 different languages (and the number is still increasing), it is questionable, whether it is possible to generate such a high number of different colour hues that the user can still distinguish.

The number of corpora – currently about 200 – and the number of different colours can be considered quite overwhelming at the first sight. However, when the user interacts with the system and filters the list, the different sizes of corpora are immediately easy to spot and compare also between languages.

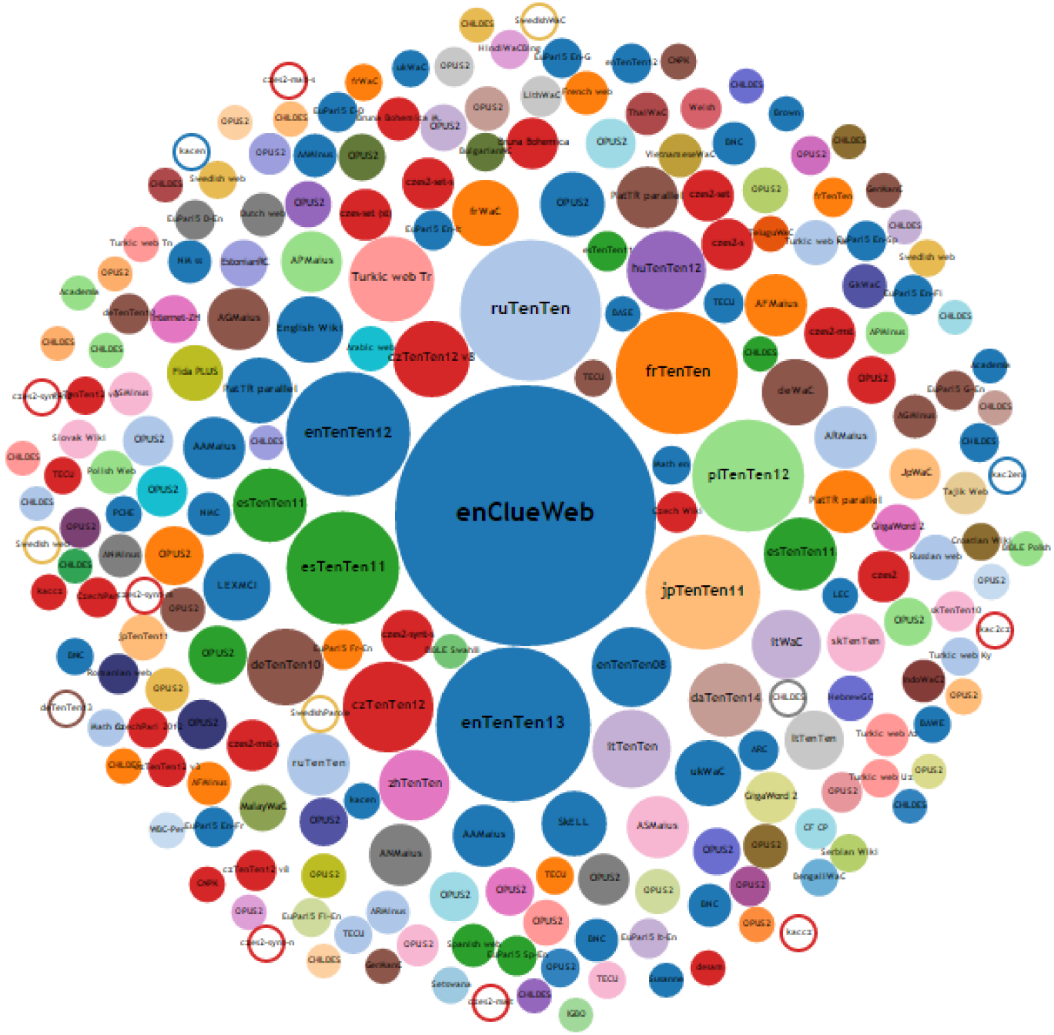


Figure 5.2: Corpora visualization

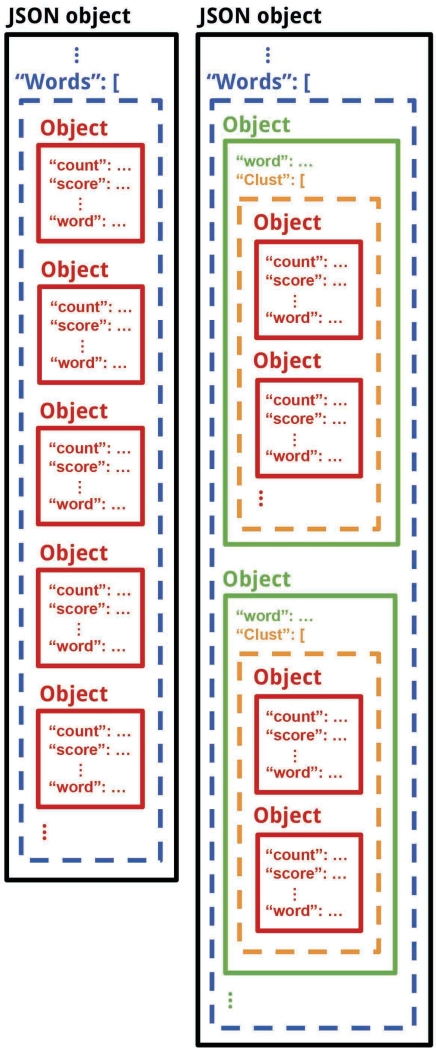


Figure 5.3: Two types of input formats for thesaurus function

The final appearance of the thesaurus visualization can be seen in Figure 5.4 which displays the output for the word "research".

The input for the thesaurus function can have two formats, as can be seen in Figure 5.3. The standard input is a simple array, where each object corresponds to one word and it has several attributes, such as count, score, id, or word.

If the clustering is enabled, each object has an additional array named Clust that contains words that belong to the given cluster. In order to use only one function for processing both JSON types, the internal structures were united – all words are placed into one array and are equipped with an attribute with the value of the id of a cluster they belong to.

The clusters are displayed as a pie chart with coloured background to distinguish them. The angles of clusters are calculated according to the overall frequency automatically with the d3.layout.pie() function.

Hovering over a word or its frequency circle displays a tooltip with the exact values, so the user can always get the precise numbers. Moreover, when hovering over the score axis, only those words that are near the particular score value remain coloured, others are displayed as gray until the mouse moves out.

To change a query, user can either type a word into the input box or click on a word that is already shown in the visualization.

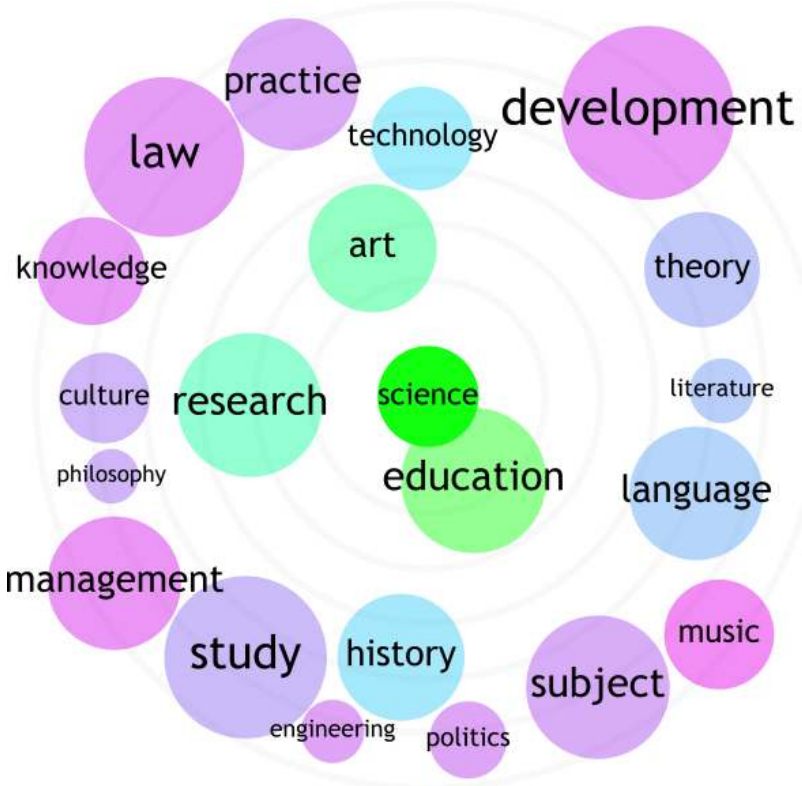


Figure 5.4: Thesaurus visualization for the word “research”

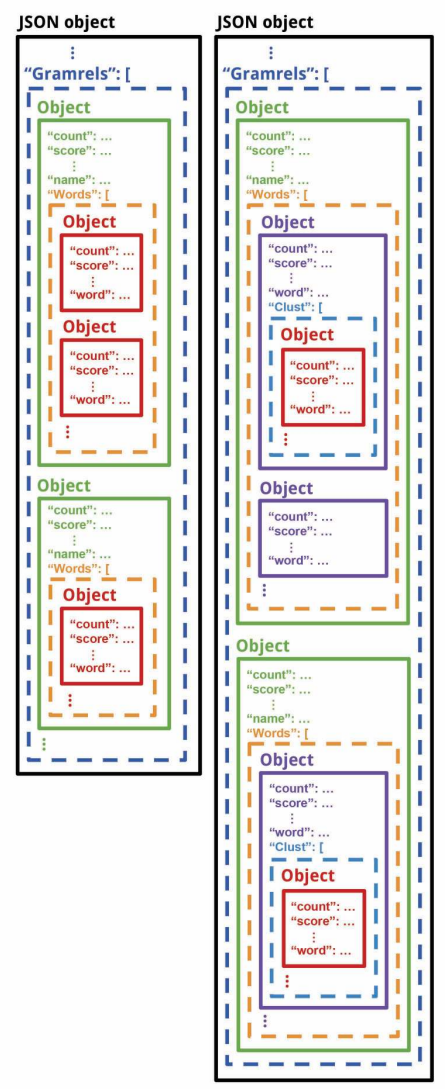


Figure 5.5: Two types of input formats for word sketch function

Figure 5.6 shows the final output of the word sketch function for the word "drop" and its three selected relations – objects, subjects, and modifiers.

Similarly as in the thesaurus, the clustering is also available for the word sketch function. Though, it results in more complex structure (see Figure 5.7) which had to be also united.

The internal structure for working with words of the word sketch is an array of objects which represents a relation. Each relation contains words with the same attributes as the thesaurus function.

In case of clustering, each cluster is parsed as one relation. This enables the user to see the different proportions of scores and frequencies.

Arcs display two information about each relation – the overall score and the overall frequency. Because the overall score was in the time of the visualization implementation being changed, the score was calculated as the average score in a given relation (but will be easily changed on the deployment in SkE). The frequency is mapped to the relations' area.

Interactions in word sketch are triggered by the same actions as in thesaurus, so users can expect the same behaviour of the system.

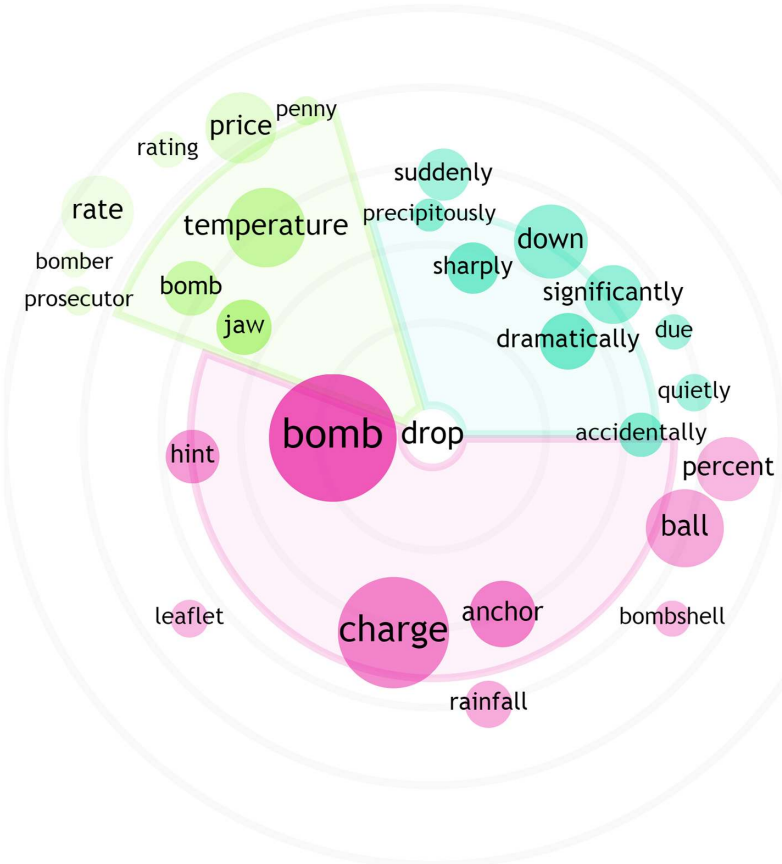


Figure 5.6: Word Sketch visualization
for the word “drop”

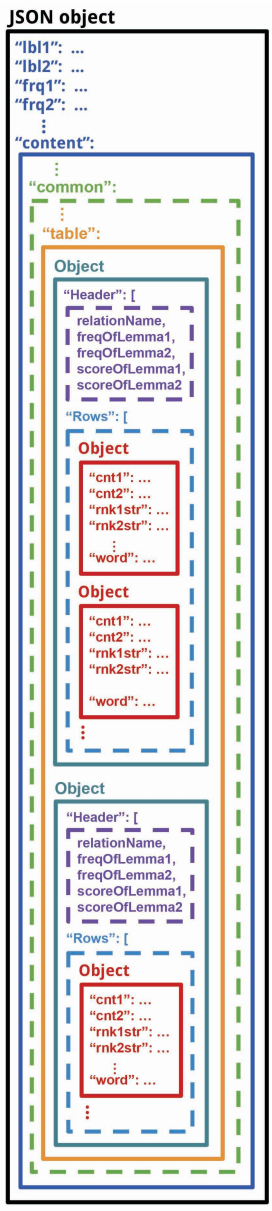


Figure 5.7: Two types of input formats for word sketch function

5.5 *Word Sketch Difference implementation*

Selected relations displayed by the word sketch visualization can be seen in Figure 5.8, which shows nouns modified by various pairs of words.

The word sketch difference function produces a number of separated visualizations for each function. As can be seen in Figure 5.7, each relation is input as an object that has general information stored in an array of the header attribute and words are passed in an array as rows. There was only a small number of changes needed to store the relations in an internal structure, for example detecting whether the score is number or renaming the attributes of words.

Relation boxes can be toggled, so the user can show or hide different visualizations and therefore focus only on those of interest. Also, the boxes were implemented using jQuery UI function .sortable() in order to ease the comparison of different relations, especially in cases where there are more than a couple of relations which would lead to enormous scrolling.

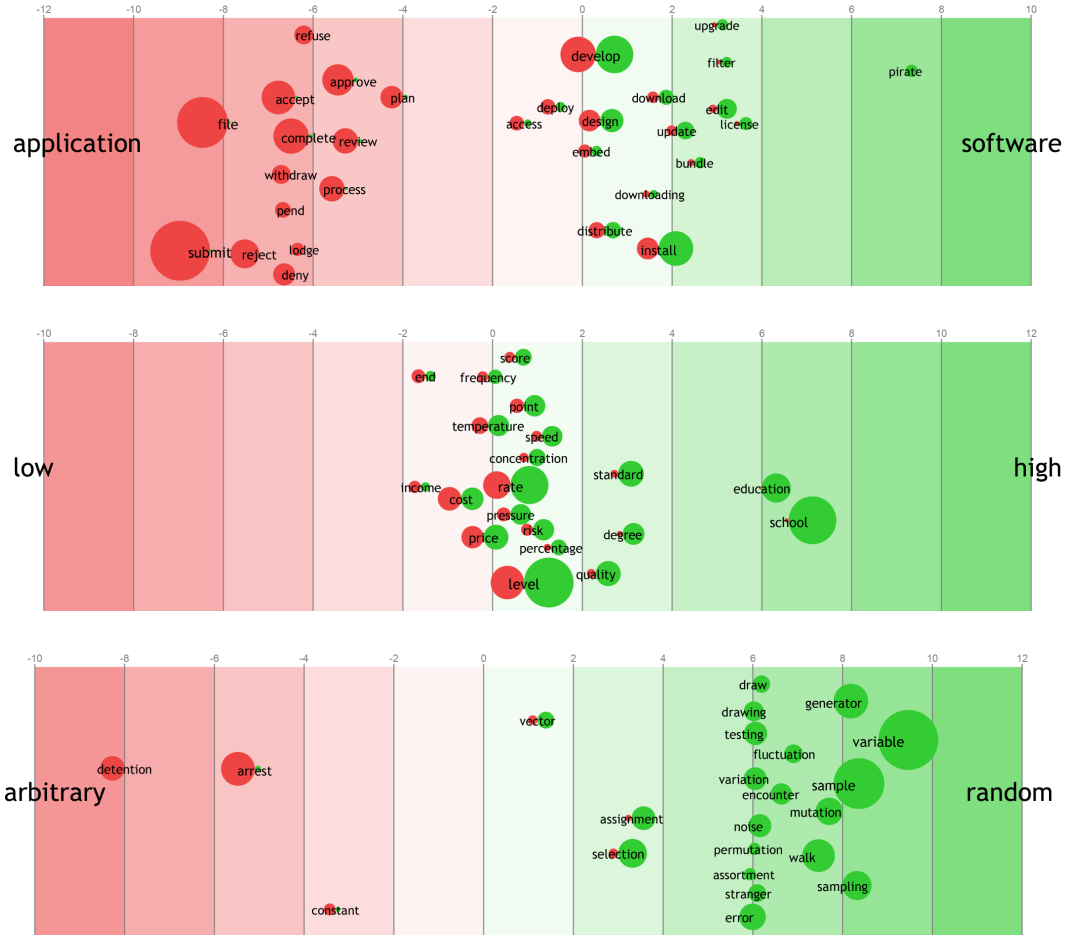


Figure 5.8: Various outputs from WS-Diff visualization

5.6 User interfaces

The user interfaces were designed to be simple and consistent as much as possible. Into the demo pages, icons were also created for increasing recognition of the features (Johnson, 2010). Their appearance was designed to illustrate a simplified version of the proposed visualizations, so the user can easily remember which output can be achieved when returning to the system (Johnson, 2010). The icons can be seen in Figure 5.9.

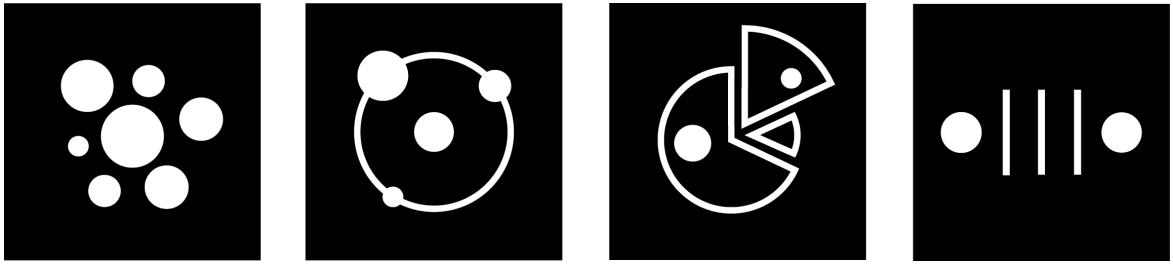


Figure 5.9: Icons designed for the demo interface

The web pages of the demo interfaces are not responsive but the future versions could be easily adapted after the viewport dimensions would be detected and attributes of SVG changed accordingly.

The interface of the corpora overview, displayed in the Figure 5.10, consists of three different tabs, each containing the visualization in the middle, the language panel on the right and the the rest of the controls are located at the bottom.

In Figure 5.11, the interface of the thesaurus function can be observed. In the top there is a question stated in order to help the users – and especially the new ones – so they know what kind of goal they can achieve with the given visualization. On the left, there is a legend, so the scales can be interpreted

correctly and also the values could be deducted. When the clustering option is set on, above the legend there is a list of clusters, where each item has a colour code assigned next to the headword. In the middle, there is the visualization and on the right there is the panel with various control options that help the users to explore the results deeper also via the interactions.

As can be seen in Figure 5.12, the interface of the word sketch function is very similar to the thesaurus function because the same placement of the same features increases the ease of learning and recognition. Moreover, the users expect to find the options at the same place, so they do not have to think about or look for the particular settings for altering the output and they can interact with the system automatically without conscious monitoring. (Johnson, 2010)

The main difference is that above the legend there is a list of relations that the user can choose from. Only the selected relations in the list have their background coloured, so distinguishing between different hues in the visualization and in the list is easy.

The word sketch function, which interface can be seen in Figure 5.13, has the control panel and the legend placed below the question because the visualization is not a square as it was in the previous examples but a rectangle which has the width of the page. Therefore, if the control panel would be placed on the right or on the left, the area for the graphics would be needlessly reduced. And because there can be a large number of relations, the controls could not be placed below the last visualization as the user would have to scroll too many times.

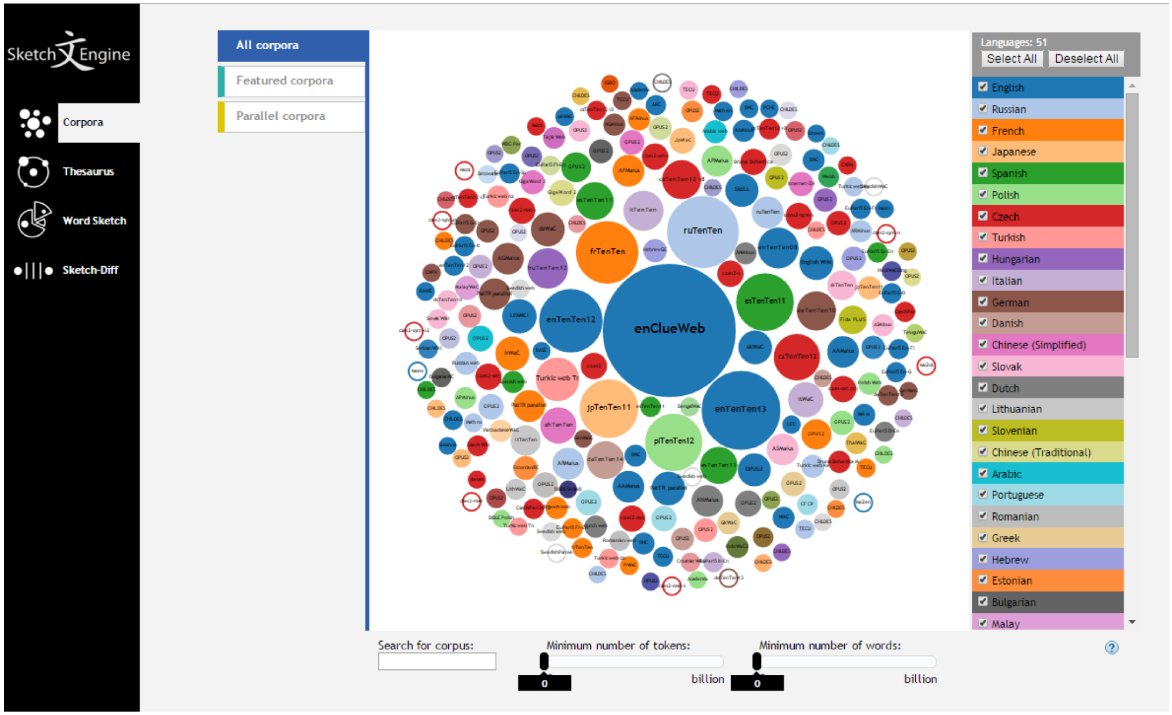


Figure 5.10: The interface of the corpora overview

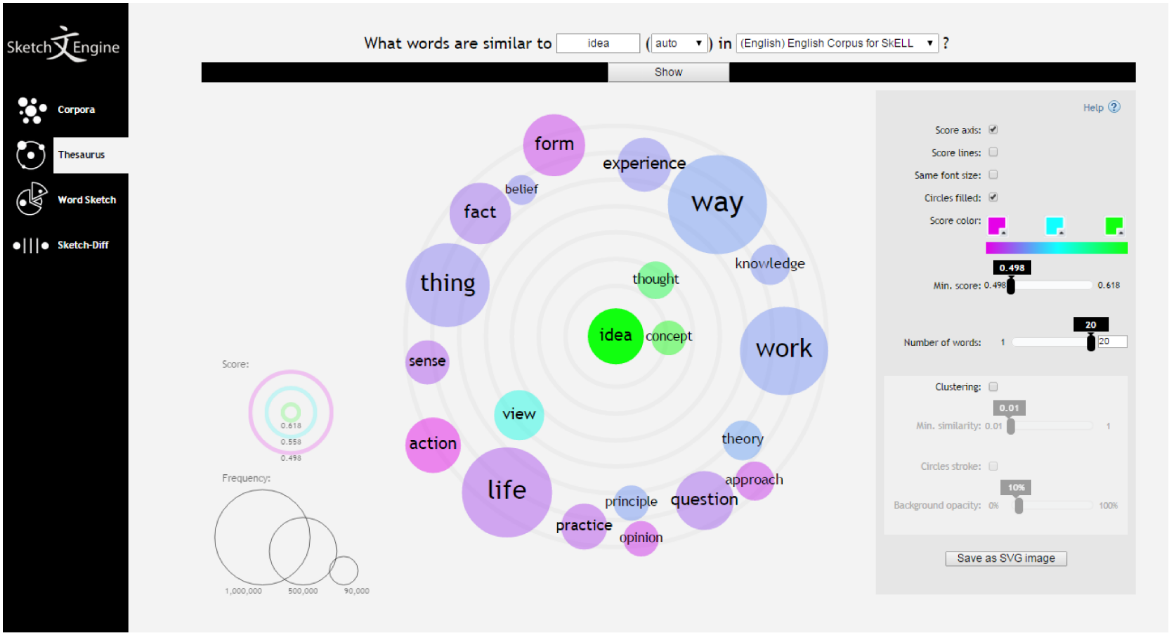


Figure 5.11: The interface of the thesaurus

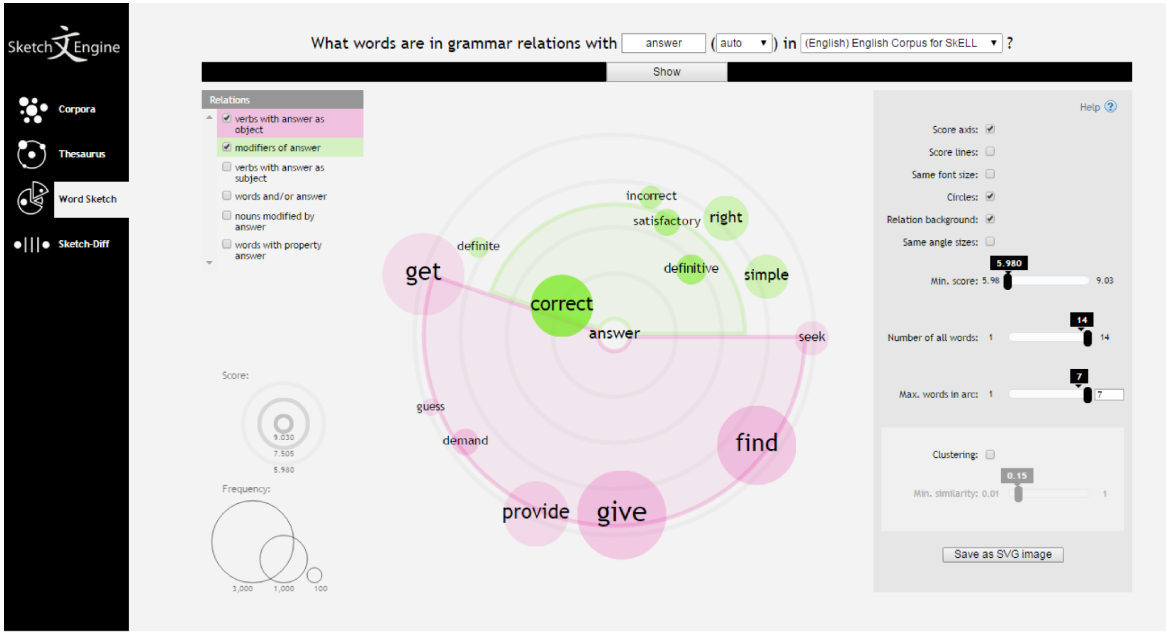


Figure 5.12: The interface of the word sketch

Figure 5.13: The interface of the word sketch difference

6
Evaluation

The purpose of the following summative evaluation is to test the implemented visualizations with users. The obtained feedback will show whether the visualizations help to easily interpret the data of SkE and will determine what attributes should be modified before the final release. In addition, the testing will reveal what features should be added in the further development.

6.1 Problem statement and test objectives

The test will provide quantitative and qualitative data from test participants that will help to determine exactly what features and to what extent are beneficial. The ability to successfully finish a typical task when using the system will be also tested with various level of expertise to find out whether the system can easily support users' goals. Moreover, the overall user experience and feeling of the system will be evaluated, so the system is not only practical but also enjoyable. Users will be requested for their comments and requirements for further development, so their opinions are taken into account before the visualizations will be officially released.

Test participants will be given an online form that consists of pre-test questions that will provide the information about their expertise of using SkE. Short tasks and post-tasks questions will follow which will provide insights about the users' workflow and the ability of the system to respond to users' expectations. Post-test questions will be given for each tool, so the overall usability and satisfiability can be also compared.

Each test participant will be scheduled an one-hour session. The only exception were expert users because they were asked to use the visualizations while working, therefore observation of them would be difficult and intimidating. The sessions will be held in conditions where anyone else will have access to ensure that the test participants will not be watched over or anyhow disturbed. The room was equipped with two monitors so the answering of questions after interacting with the system would be easier and the user's attention would not be intruded.

6.2 User profiles of SkE

Establishing subgroups of users according to the level of experience with a given software is often one of the most difficult perspectives to pick upon (Dumas and Redish, 1999); however the boundaries between different knowledge of SkE users were set to be intelligible enough to clearly identify the particular subgroup:

- new users – people who have never used SkE before,
- intermediate users – people who have some knowledge of the system and used it before or use it sometimes but have no detailed knowledge of additional functions,
- expert users – people who use SkE in their daily job and know its features in detail,
- expert users (developers) – people who are involved in the design and development of SkE.

Involving developers and creating a subgroup for them can yield interesting results about different perspectives and expectations of the system. They could be a part of other subgroup but because of their involvement in the process of the system development, they have distinct and detailed knowledge as no other users do.

Users with different backgrounds and variety of careers were recruited as test participants – varying from computer linguistics through information studies to economics, which corresponds to the distinctive user base of SkE.

6.3 Methodology and tasks

The total number of participants was 23, where in each group there were at least 5 participants. This number is higher than the minimum that it is recommended and usually practised, as described in (Krug, 2009), (Dumas and Redish, 1999), (Barnum,

2010), (Virzi, 1992) because the visualizations are supposed to be used by a variety of users with different background, so it will be possible to see if they will bring distinct or the same findings and remarks.

The think aloud process will be used only for the first part of the session because when performing tasks the first experience and reactions to the interactions and the system as the whole is crucial, so any interruption could skew the user's responds. However, additional testing can be performed when the results could not be interpreted explicitly or will show that the response could be measured in means of performance.

The method for user testing in person was moderated and remote user testing was unmoderated. To be able to follow users' actions and analyse them even in the case of unmoderated testing, tracking of the webpages was set up using Open Web Analytics [1] and Usability Tools [2]. Detailed description of these tools are in the next section 6.4.

[1] http://www.openwebanalytics.com/
[2] https://usabilitytools.com/

There will be two types of data collected – qualitative and quantitative – using questionnaires. Qualitative data will be represented by notes from observation, comments by users while testing and written feedback after the testing. Quantitative data will be gained only from questionnaires – users will provide preferences regarding the usefulness and usability.

The overview of the session progress

- Briefing – 1 min
- For each of the four proposed visualizations the main testing comprised of:
  1. Overview of the system and pre-test questionnaire – 5 mins
  2. Tasks and post-tasks questionnaires – 3 mins
  3. Post-test questionnaire – 2 mins
- Closing – 1 min

Each task was designed to set a basic goal that can be accomplished with the system so the way to achieve the goal can be compared. It will also determine the different approaches that users followed to complete each task, therefore the feedback for user interface will be given.

The questionnaires were designed to be as short as possible, due to the fact that there are four types of visualizations that needed to be rated by very similar questions, but also as informative as needed. Moreover, there are all three types of questionnaires, so the system can be evaluated from different usability points.

Each questionnaire for a given visualization includes two sets of Likert-type items to measure user preferences of controls and overall usability. The Likert-type item is a question or a statement that is followed by response options that a participant chooses from. Four Likert-type items can be seen in Figure 6.1. The number of response options may vary as well as the labels given to them, so to set a proper label that would not lead to any misinterpretation and which would be symmetric, the 7-point Likert-type anchors were adapted according to (Vagias, 2006). These seven options were chosen because as shown in (Finstad, 2010), they allow to measure more accurately the responses in contrast to five options when using electronic tools, such as Google Form which was used to collect the feedback in this thesis.

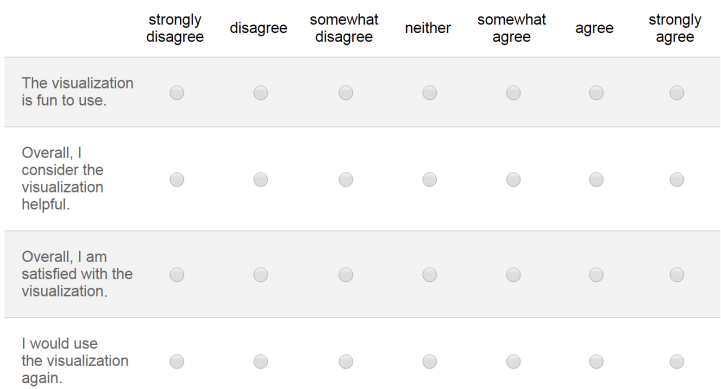


Figure 6.1: Two types of input formats for thesaurus function

Likert-type item originates from Likert scale which was designed to measure a person's attitude towards a given issue. Likert scale is build from several Likert-type items that are summed into a single number which is then interpreted to quantitatively measure the attitude of a given correspondent. (Boone and Boone, 2012) Because the first set of Likert-type items measures the usefulness of each control component, it is desirable to interpret them only as single questions. The second set, which examines the overall usability attitudes, could be interpreted as only a single item. However, the same statements regarding different visualizations will be evaluated, therefore even in this case a Likert scale was not applied.

Full questionnaires can be found in the attachment file *testing_questionnaire.pdf*.

6.4 Website analytics tools

Quantitative analysis can be extended also with automatically collected data, which can show the usability statistics that users are not aware of while using a system. In systems which are using web interface, the mostly used approach in gathering and analysing data is through a web analytics service, such as Google Analytics. The main advantage of these systems is that they are installed on a server, so the user is not obliged to install any additional software to his or her computer. In addition, these tools offers also reporting services which summarize the stored data and run through various statistics that reveal users behavior.

Well-known web analytics provide the basic features, such as user uniqueness or page count. However, when testing of corresponding visualizations, there needed to be more proficient tools for page analysis, for example heat map generation and

mouse tracking. Whereas several open source tools for heat maps are available, such as clickheat[3], open source applications for mouse tracking are hard to find; most available tools are paid or they offer a freemium[4] package which has a restricted functionality.

When searching for the most suitable tool, various services were installed and tried in the local installation:

- MouseStats[5]
- DecibelInsight[6]
- SessionCam[7]
- Inspectlet[8]
- UsabilityTools[9]
- OpenWebAnalytics[10]

First three tools, MouseStats, Decibel Insight, SessionCam, were not able to interpret the mouse movements correctly – either the mouse replay was misplaced or the heat map was generated incorrectly. The tools could not handle the dynamic content of the webpage, so they were excluded from the system and were not used for user testing.

Inspectlet tracked and interpreted the data mostly correctly; hovewer, there were only four sessions a day available in the freemium package, which was not sufficient for the testing because each day there could created at least 5 different sessions.

Usability tools, a paid tool with free 14-day trial was the best alternative from freemium services – it correctly interpreted gathered data and interpreted it almost always correctly – the only thing not working as same as in the life page session were regenerating some elements when replaying the user's session. Nevertheless, it offered at least the correct mouse movements over the whole page and advanced heat maps. This system was integrated during the user testing and the outputs are available in the section 6.5.

[3] labsmedia.com/clickheat

[4] service or product is free of charge but enabling additional functions demands payment

[5] http://www.mousestats.com/

[6] https://usabilitytools.com

[7] https://usabilitytools.com

[8] https://usabilitytools.com

[9] https://usabilitytools.com

[10] http://www.openwebanalytics.com

Open Web Analytics, later mentioned as OWA, is an open source solution to web page tracking; in the time of writing this thesis, the current version was 1.5.7. Because it is not an external service, the whole CMS system with a database was required to set up on the server hosting web interface of the visualizations. Besides providing common features of web analytics tools, such as user or page statistics, it also provides advanced mouse tracking. Data from mouse tracking can be automatically converted into a heat map, an animation of user visit and statistics about clicking on various DOM elements. As other tools, it could not interpret correct mouse movements and update of page elements at the same time; however, the mouse stream could be used in some cases. This system was also included for user testing.

Many features of web analytics software, such as the number of returning users or detailed statistics of pages, can be used after the visualizations will be incorporated in the released version of SkE. Also, relevant information will be collected after a longer period of time and by more users which is beyond the scope of this thesis.

6.5 Results of user testing

Users have found the visualizations very eye-catching, especially the ones that have used SkE before. Because in the questionnaire there were ten statements about the overall usability to each proposed visualization, this chapter will consider only the most important ones and the rest can be found in the attached document.

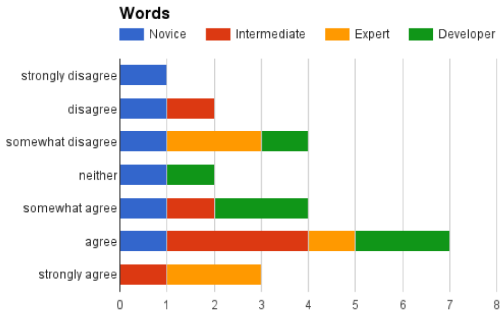
The most crucial statement that reveals whether the visualizations helped the users to achieve their goals is summarized in “I would use the visualization again.”. Another one that will

be considered is "Overall, I consider the visualization helpful.", as it shows whether the visualization is suitable for the users' tasks.

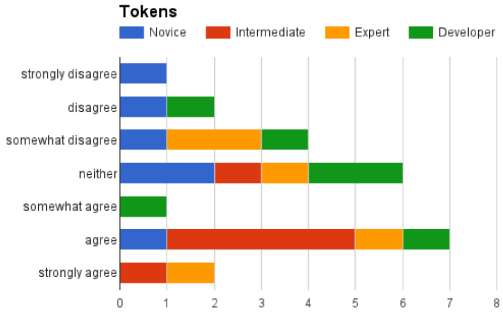
6.5.1 Corpora overview

Corpora visualization was considered really appealing and very fun to use but the users were confused about the terminology, specifically among tokens and words, as was expected. However, the users did not agree on which of those two terms they prefer, not even amongst different profiles as can be seen in Figure 6.2. But they agreed on that they would use only one of them if it would be the only option. Therefore, they were quite confused why the interface had two sliders and were not sure how they are updating each other automatically.

Other controls were found very useful, especially the language select option, but the users did not understand the ordering in the list. They would expect an alphabetical order or ordering by language families.

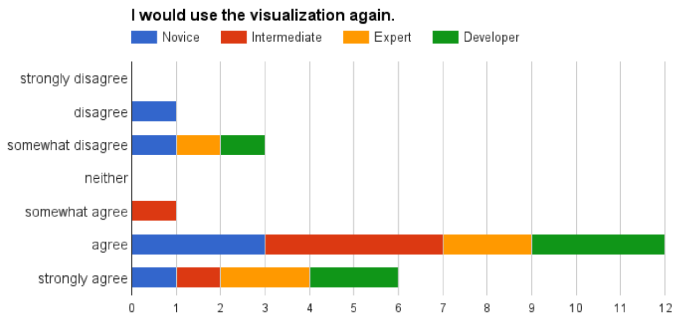


(a) words

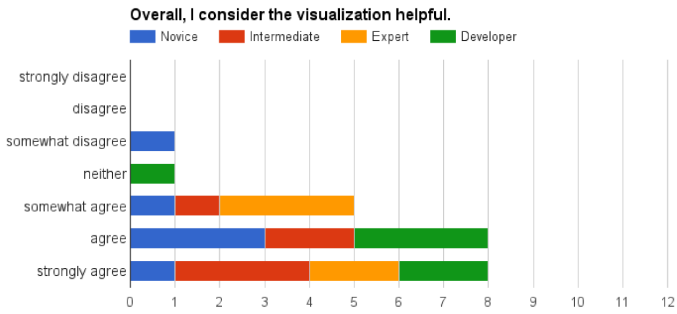


(b) tokens

Figure 6.2: Corpora visualization evaluated



(a) Overall, I consider the visualization helpful.



(b) I would use the visualization again.

Figure 6.3: Corpora visualization evaluated

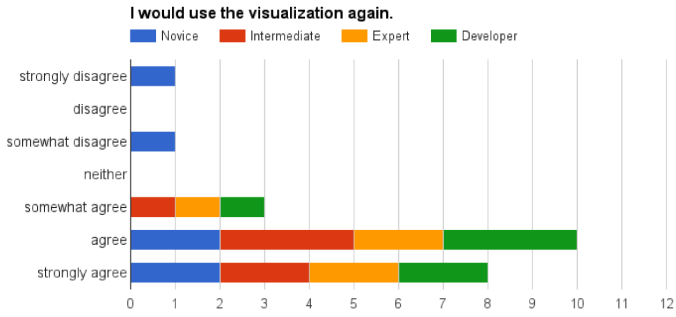
The mapping of data attributes was generally correctly understood. The only attribute the users did not understand well or were not sure about was the font size.

The tasks were very easy to successfully finish as only two users made an error while looking for a number of corpora family that had "Maius" in the name and only one made a mistake when determining the biggest German corpus.

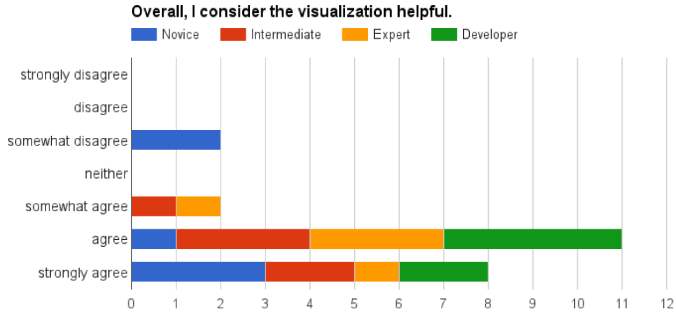
Overall, most of the users would use the visualization again, as can be seen in Figure 6.3 and also slightly over two thirds considered the visualization helpful.

6.5.2 *Thesaurus*

The Thesaurus was considered by the majority of users easy to understand and quite helpful. The users were concentrating on the positions of words, so they had no problems with identifying the mapping of different attributes. The score colours were mostly not noticed until the users reached out to change them. They were firstly looking surprised at the output but after they tried to change a colour again, their confidence about its meaning and the whole output was boosted. However, there were two novice users who were confused by some of its features, as can be seen in Figure 6.4, especially by the meaning of some of the labels and what a score represents. Therefore, these issues will need to be resolved in order to provide more clear and easy-to-use tool for analyzing similar words.



(a) Overall, I consider the visualization helpful.



(b) I would use the visualization again.

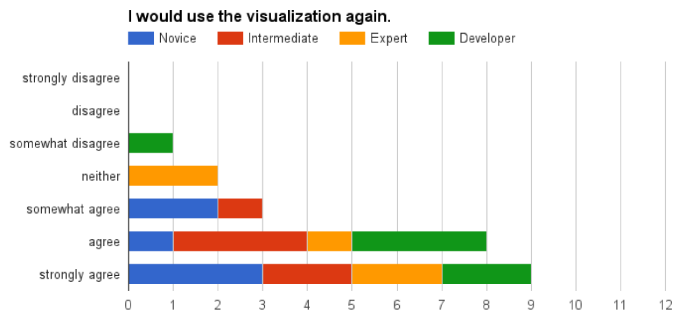
Figure 6.4: Thesaurus visualization evaluated

The clustering option appreciated only a fraction of the users as they did not use this feature in SkE before. Additionally, they did not fully understood the meaning of minimum similarity slider. This was partially caused by wrong implementation of its slider – by moving the slider with mouse pressed, all values between the original and the final state of the slider were requested, so the response was very slow and the image was changing rapidly and unexpectedly.

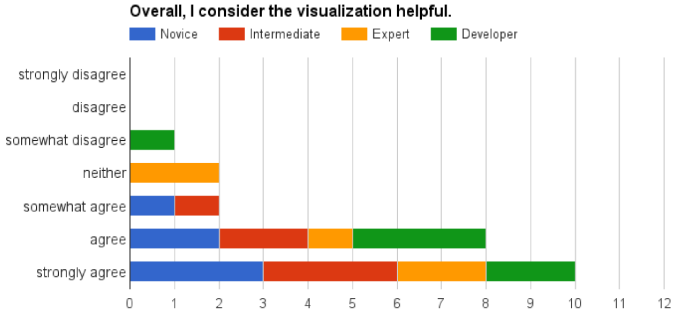
6.5.3 *Word Sketch*

The Word Sketch visualization was considered to be very useful in overall users' evaluation, as can be seen in Figure 6.5. They were satisfied with the outputs and correctly identified the mapping combinations of data and graphical elements. Therefore, they had no problems with finishing the tasks quickly and correctly.

Figure 6.5: Word Sketch visualization evaluated



(a) Overall, I consider the visualization helpful.



(b) I would use the visualization again.

The only issue the users were experiencing was when working with relations which had too small frequency values and therefore the words within were overlapping without any option to reveal the words in a readable form. This situation is depicted in Figure 6.6. The users suggested to show the words with a white background on hover or increasing the angle of the relation on hover.

As in thesaurus, the minimal similarity slider for clustering option was not implemented properly and was causing users to change their way of interacting with the interface.

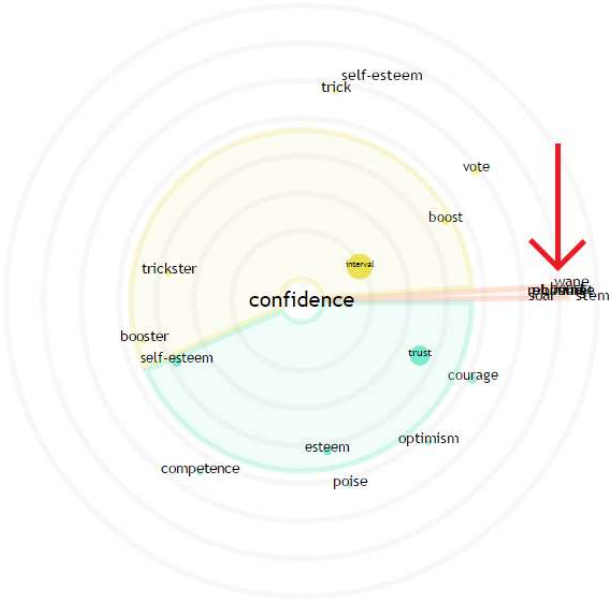


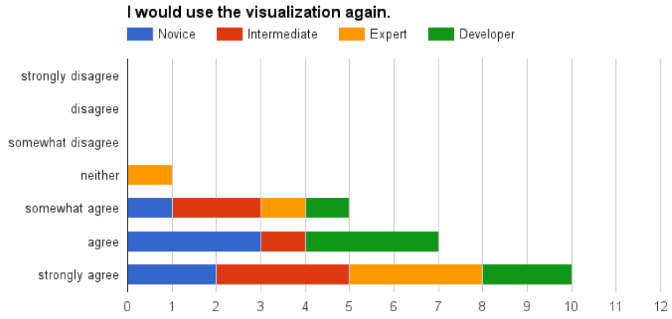
Figure 6.6: An example of Word Sketch with unsatisfying output

6.5.4 *Word Sketch Difference*

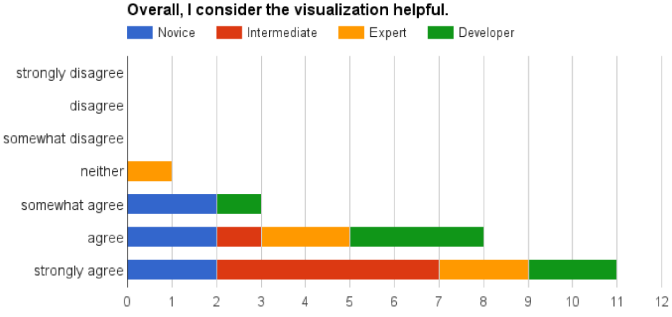
The Word Sketch Difference visualization was considered by the users as the most useful one, as can be seen in Figure 6.7. The users understood correctly all attributes that were mapped and all except one stated that the visualization was helpful while doing a task – they usually noted that "It is obvious" when filling the questionnaires.

Displaying of score lines were not really preferred and if it was, the users were not satisfied with it because some of the lines were overlapping. In order to correctly and easily tell apart the lines, the users proposed an animation of word's frequency circles or increasing the height of the given panel when the option would be selected.

The tiled score axis gradient was the best satisfying for the users and the purpose of the minimal frequency option was not clear even to intermediate or some of the expert users.



(a) Overall, I consider the visualization helpful.



(b) I would use the visualization again.

Figure 6.7: Word Sketch Difference visualization evaluated

6.6 User testing evaluation

The user testing revealed that the users would prefer to have only a very few options in the control panels. Therefore, the options would be reduced in the release version. There would be also some small but very important implementation changes that will cause much more user friendly behaviour. For example, the input fields were not creating a new query when the enter button was clicked and the clustering slider for minimum similarity was not properly implemented as mentioned above.

The tools used for tracking the usage of visualizations will reveal only a small glimpse on how users interact with the site and they will surely not answer "why" questions. However, they will show what should be tested in the next user testing sessions to get the answers on the former issues. Figure 6.8 demonstrates one of the possible results of a usability tool – hover heatmap. Figure clearly shows the areas where the user hovered or stayed with the mouse cursor for most of the time. Therefore, identifying the areas of interests or unexpected behaviour can be easily seen.

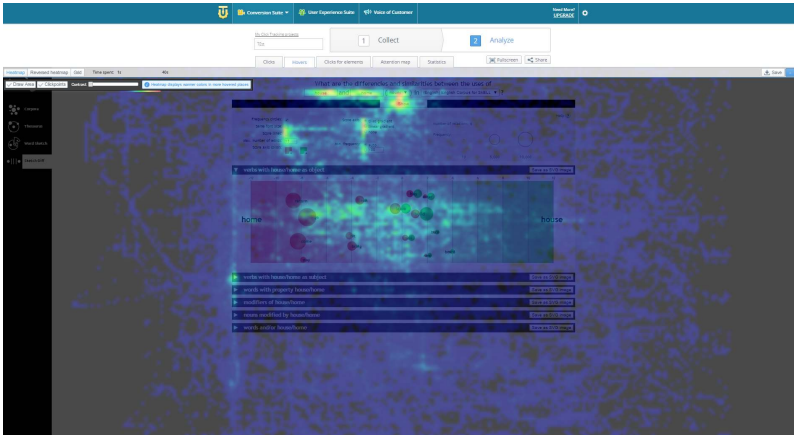


Figure 6.8: Hover heatmap for the Word Sketch Difference, generated by UsabilityTools

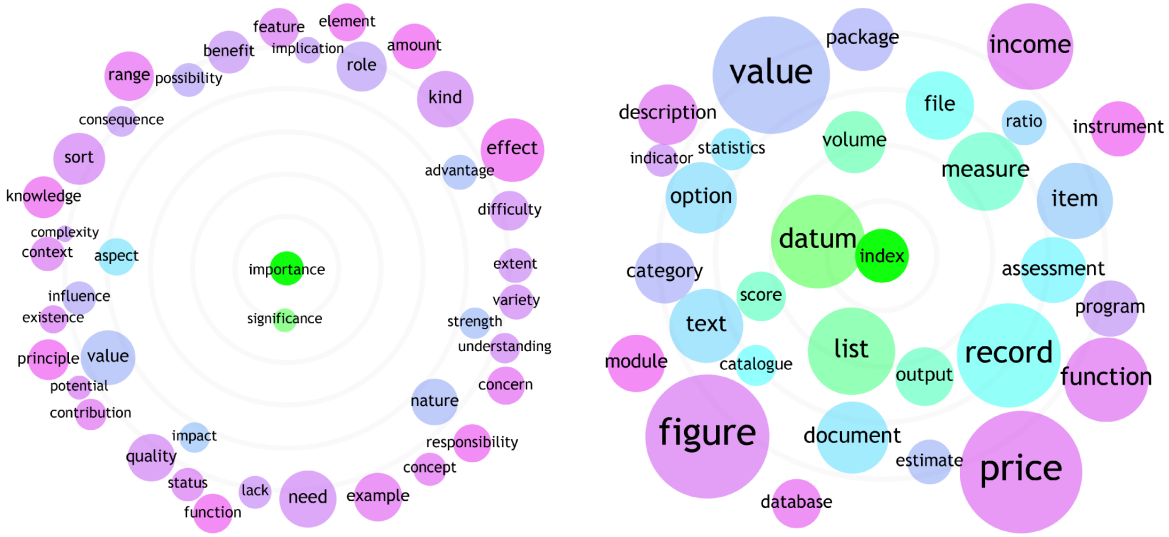
7
Conclusion

Interactive data visualization can bring a new way of seeing and exploring data but the more complex data set we have, the more we have to think about how the particular parts of data will be mapped in the final visualization, so the user has no or little opportunity to misinterpret it. When a concept is used in a correct way, it can impress and convey interesting and important knowledge at the same time. Moreover, it can make our everyday stereotypical routines easier. Therefore, data visualization is not only the graphic art but it can change the way we perceive information or additionally improve our lives.

The main objective of this thesis was to design a number of concepts for the core functions of the Sketch Engine. The concepts were discussed with the developers of the system in order to enable the users to achieve their goals. Selected designs were successfully implemented as an online demo that is easy to embed not only in the SkE interface but also in any other website.

Subsequent user testing revealed that the visualizations were useful not only for new but also for intermediate users. The expert users confirmed that the visualizations would be useful as an additional representation of data in the SkE interface.

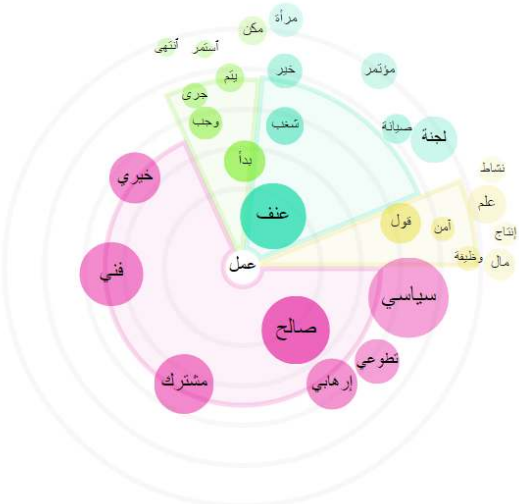
The main advantage of using the visualization for thesaurus is that it reveals whether the query has only a few close words, such as importance shown in Figure 7.1 (a), or the words are used in richer or more meaningful contexts, such as index in Figure 7.1 (b).



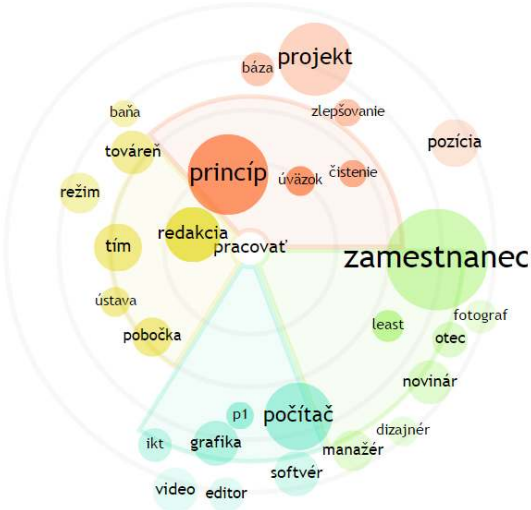
(a) importance **(b)** index

Figure 7.1: Different thesaurus outputs

Because the implemented scripts do not work with words as strings but only as elements that need to scaled and placed, all the visualizations are language independent and therefore usable with any corpus that is able to provide thesaurus, word sketch or word sketch difference from its data. Figure 7.2 illustrates different outputs in Slovak and Arabic for the same query.



(a) Arabic corpus



(b) Slovak corpus

Figure 7.2: Different word sketch of "work" from

Partial outputs of the thesis, specifically thesaurus and word sketch visualizations, were presented in QueryVis - Workshop on Innovative Corpus Query and Visualization Tools at *20th Nordic Conference of Computational Linguistics 2015* (Kocincová et al., 2015) with positive receptions.

Although the visualizations will be incorporated in SkE along with the tabled data, they can be used also independently as well as they do not require any knowledge of the data or the calculation process behind.

7.1 Future work

In the nearest future, the proposed visualizations will be firstly revised, so the user feedback will be reflected in the implementation and then they will be deployed into the SkE online interface. The user feedback will be continuously collected and evaluated in order to determine if the features are causing any further problems. Moreover, several A/B testing will be planned to test hypotheses that came out from the users' feedback.

Because in some cases the algorithms can't fit the words properly, the scales are shrank even though it is possible to fit the elements within the image without changing the scales. Therefore, first-fit placing algorithm will be replaced by best-fit.

Additionally, the scripts will be tested whether it is desirable to make optimizations or not.

Designing and implementation of visualizations for other features of SkE is also planned, so the full potential of data within the system will be exploited.

Bibliography

Baisa, V. and Suchomel, V. (2014). Skell: Web interface for english language learning. In *Eighth Workshop on Recent Advances in Slavonic Natural Language Processing*, pages 63–70, Brno. Tribun EU.

Barnum, C. M. (2010). *Usability Testing Essentials: Ready, Set...Test!* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1st edition.

Boone, H. N. and Boone, D. A. (2012). Analyzing likert data. *Journal of Extension*, 50(2):1–5.

Börner, K. and Polley, D. (2014). *Visual Insights: A Practical Guide to Making Sense of Data*. MIT Press.

Bostock, M. (2013). Quantitative scales mbostock/d3 Wiki GitHub. https://github.com/mbostock/d3/wiki/Quantitative-Scales. Last accessed on May 10, 2015.

Bostock, M. (2015a). Home mbostock/d3 Wiki GitHub. https://github.com/mbostock/d3/wiki. Last accessed on May 10, 2015.

Bostock, M. (2015b). Selections mbostock/d3 Wiki GitHub. https://github.com/mbostock/d3/wiki/Selections. Last accessed on May 10, 2015.

Bostock, M. and Heer, J. (2009). Protovis: A Graphical Toolkit for Visualization. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1121–1128.

Bostock, M., Ogievetsky, V., and Heer, J. (2011). D3 Data-Driven Documents. *IEEE Transactions on Visualization and Computer Graphics*, pages 2301–2309.

Cambridge University Press (2015). Thesaurus definition, meaning - what is thesaurus in the british english dictionary & thesaurus - cambridge dictionaries online. http://dictionary.cambridge.org/dictionary/british/thesaurus. 2015-04-25.

Card, S. K., Mackinlay, J. D., and Shneiderman, B., editors (1999). *Readings in Information Visualization: Using Vision to Think*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Cecco, R. (2011). *Supercharged JavaScript Graphics: With HTML5 Canvas, JQuery, and More*. O'Reilly Series. O'Reilly Media, Incorporated.

Christ, O. and Schulze, B. M. (1994). *The IMS Corpus Workbench: Corpus Query Processor (CQP) Users Manual*. University of Stuttgart, Stuttgart, Germany.

Chuang, J., Manning, C. D., and Heer, J. (2012). Termite: Visualization techniques for assessing textual topic models. In *Proceedings of the International Working Conference on Advanced Visual Interfaces*, pages 74–77. ACM.

Collins, C., Viegas, F. B., and Wattenberg, M. (2009). Parallel tag clouds to explore and analyze faceted text corpora. In *Visual Analytics Science and Technology, 2009. VAST 2009. IEEE Symposium on*, pages 91–98. IEEE.

Cooper, A., Reimann, R., and Cronin, D. (2007). *About Face 3: The Essentials of Interaction Design*. John Wiley & Sons, Inc., New York, NY, USA.

Culy, C., Passarotti, M., and Knig-Cardanobile, U. (2014). A compact interactive visualization of dependency treebank query results. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, Reykjavik, Iceland. European Language Resources Association (ELRA).

Deveria, A. (2015). Can i use... support tables for html5, css3, etc. http://caniuse.com. Last accessed on May 10, 2015.

Dumas, J. S. and Redish, J. C. (1999). *A Practical Guide to Usability Testing*. Intellect Books, 1st edition.

Dunn, P. (2007). Visuwords - online graphical dictionary. http://www.visuwords.com. Last accessed on April 25, 2015.

Fellbaum, C. (2005). Wordnet and wordnets. In Brown, K., editor, *Encyclopedia of Language and Linguistics*, pages 665–670, Oxford. Elsevier.

Ferster, B. (2012). *Interactive Visualization: Insight Through Inquiry*. The MIT Press.

Finstad, K. (2010). Response interpolation and scale sensitivity: Evidence against 5-point scales. *Journal of Usability Studies*, 5(3):104–110.

Flanagan, D. (2011). *JavaScript: The Definitive Guide*. Definitive Guides. O'Reilly Media, 6th edition.

Havre, S., Hetzler, E., Whitney, P., and Nowell, L. (2002). Themeriver: visualizing thematic changes in large document collections. *Visualization and Computer Graphics, IEEE Transactions on*, 8(1):9–20.

Hearst, M. A. (1995). Tilebars: Visualization of term distribution information in full text information access. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '95, pages 59–66, New York, NY, USA. ACM Press/Addison-Wesley Publishing Co.

Hearst, M. A. (2009). *Search User Interfaces*. Cambridge University Press, New York, NY, USA, 1st edition.

Hilbert, M. and Lpez, P. (2011). The Worlds Technological Capacity to Store, Communicate, and Compute Information. *Science*, 332(6025):60–65.

Iliinsky, N. (2010). On beauty. In Steele, J. and Iliinsky, N., editors, *Beautiful Visualization: Looking at Data Through the Eyes of Experts*. O'Reilly Media, Inc., 1st edition.

Jennifer, N. R. (2013). *HTML5: pocket reference*. OReilly Media.

Johnson, J. (2010). *Designing with the Mind in Mind: Simple Guide to Understanding User Interface Design Rules*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

jQuery Foundation - jquery.org (2015). jquery. `http://jquery.com/`. Last accessed on May 10, 2015.

Kilgarriff, A. (2010). Comparable corpora within and across languages, word frequency lists and the kelly project. In *Proceedings of the 3rd Workshop on Building and Using Comparable Corpora*, pages 1–5.

Kilgarriff, A. (2013). Terminology finding, parallel corpora and bilingual word sketches in the sketch engine. In *Proc ASLIB 35th Translating and the Computer Conference, London*.

Kilgarriff, A., Baisa, V., Bušta, J., Jakubíček, M., Kovář, V., Michelfeit, J., Rychlý, P., and Suchomel, V. (2014). The sketch engine: Ten years on. *Lexicography*, 1(1):736.

Kilgarriff, A., Husák, M., McAdam, K., Rundell, M., and Rychlỳ, P. (2008). Gdex: Automatically finding good dictionary examples in a corpus. In DeCesaris, E. B. . J., editor, *Proceedings of the XIII Euralex Congress*, Barcelona. Universitat Pompeu Fabra.

Kilgarriff, A., Kovář, V., Krek, S., Srdanović, I., Tiberius, C., et al. (2010). A quantitative evaluation of word sketches. In *Proceedings of the 14th EURALEX International Congress, Leeuwarden, The Netherlands*.

Kilgarriff, A., Rychly, P., Smrz, P., and Tugwell, D. (2004). The sketch engine. In *Proceedings of EURALEX*.

Kocincová, L., Baisa, V., Jakubíček, M., and Kovář, V. (2015). Interactive Visualizations of Corpus Data in Sketch Engine. In *Proceedings of the Workshop on Innovative Corpus Query and Visualization Tools at NODALIDA 2015*, Vilnius, Lithuania.

Krug, S. (2009). *Rocket Surgery Made Easy: The Do-It-Yourself Guide to Finding and Fixing Usability Problems*. New Riders Publishing, 1st edition.

Lexical Computing Ltd (2010). Clustering the "similar words" and collocates in sketch engine. https://www.sketchengine.co.uk/documentation/wiki/SkE/ClusteringNeighbours. Last accessed on April 25, 2015.

Lyding, V., Nicolas, L., and Stemle, E. (2014). 'interhist' - an interactive visual interface for corpus exploration. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, Reykjavik, Iceland. European Language Resources Association (ELRA).

McEnery, T. and Wilson, A. (1996). *Corpus linguistics*. Edinburgh University Press.

McFarland, D. S. (2013). *CSS3: The Missing Manual*. O'Reilly Media, Inc., 3rd edition.

Meirelles, I. (2013). *Design for Information: An Introduction to the Histories, Theories, and Best Practices Behind Effective Information Visualizations*. Rockport Publishers.

Microsoft Developer Network (2015). Svg vs canvas: how to choose. https://msdn.microsoft.com/en-us/library/gg193983(v=vs.85).aspx. Last accessed on May 10, 2015.

Murray, S. (2013). *Interactive Data Visualization for the Web*. O'Reilly Media, Inc.

Paley, W. B. (2002). Textarc: Showing word frequency and distribution in text. *InfoVis*.

Paranyushkin, D. (2011). Identifying the pathways for meaning circulation using text network analysis. *Nodus Labs, Berlin*.

Reiterer, H., Tullius, G., and Mann, T. M. (2005). Insyder: a content-based visual-information-seeking system for the web. *International Journal on Digital Libraries*, 5(1):25–41.

Rembold, M. and Spth, J. (2001). Munterbund. http://www.munterbund.de/visualisierung_textaehnlichkeiten/essay.php. 2015-04-25.

Ruecker, S., Radzikowska, M., Michura, P., Fiorentino, C., and Clement, T. (2009). Visualizing repetition in text. *Digital Studies / Le champ numrique*, 1(3).

Rychlý, P. (2007). Manatee/bonito - a modular corpus manager. In *1st Workshop on Recent Advances in Slavonic Natural Language Processing*, pages 65–70, Brno. Masarykova univerzita.

Rychlý, P. (2008). A lexicographer-friendly association score. In *RASLAN 2008*, pages 6–9. Masarykova Univerzita.

Rychlý, P. and Kilgarriff, A. (2007). An efficient algorithm for building a distributional thesaurus (and other sketch engine developments). In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, ACL '07, pages 41–44, Stroudsburg, PA, USA. Association for Computational Linguistics.

Snow, J. (1855). *On the Mode of Communication of Cholera*. John Churchill.

Steele, J. and Iliinsky, N. (2010). *Beautiful Visualization: Looking at Data Through the Eyes of Experts*. O'Reilly Media, Inc., 1st edition.

Sucan, M. (2010). Svg or canvas? hoosing between the two. https://dev.opera.com/articles/svg-or-canvas-choose/. Last accessed on May 10, 2015.

The World Wide Web Consortium (W3C) (2015a). HTML5 - Canvas element. http://www.w3.org/TR/html/scripting-1.html#the-canvas-element. 2015-05-08.

The World Wide Web Consortium (W3C) (2015b). SVG - about SVG. http://www.w3.org/TR/SVG/intro.html. Last accessed on April 08, 2015.

Thomas, J. (2015). *Discovering English with Sketch Engine*. Versatile.

Vagias, W. M. (2006). Likert-type scale response anchors. *Clemson International Institute for Tourism & Research Development, Department of Parks, Recreation and Tourism Management, Clemson University*, 257(6):50–58.

Van Den Broek, P. (1995). A 'landscape' model of reading comprehension: Inferential processes and the construction of a stable memory representation. *Canadian Psychology/Psychologie canadienne*, 36(1):53.

van Ham, F., Wattenberg, M., and Viegas, F. B. (2009). Mapping text with phrase nets. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1169–1176.

Viegas, F. B., Wattenberg, M., and Feinberg, J. (2009). Participatory visualization with wordle. *IEEE Transactions on Visualization and Computer Graphics*.

Virzi, R. A. (1992). Refining the Test Phase of Usability Evaluation: How Many Subjects is Enough? *Hum. Factors*, pages 457–468.

Wattenberg, M. and Viegas, F. (2008). The word tree, an interactive visual concordance. *Visualization and Computer Graphics, IEEE Transactions on*, 14(6):1221–1228.

Wikimedia Commons (2015). Anscombe's quartet. http://en.wikipedia.org/w/index.php?title=File:Anscombe%27s_quartet_3.svg&page=1. Last accessed on April 10, 2015.

ANNEX I

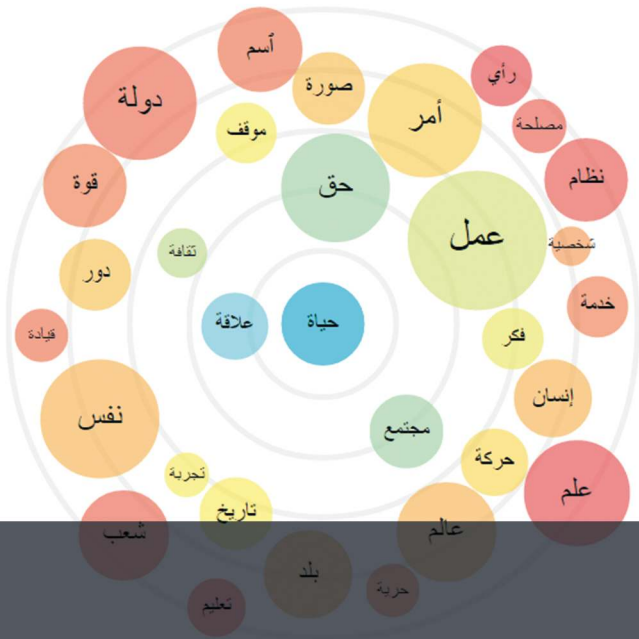
List of attached files:

- Scripts:
  - vis_main.js
  - vis_corp.js
  - vis_thes.js
  - vis_sketch.js
  - vis_wsdiff.js
- Stylesheets:
  - vis.css
- Templates:
  - corp_info.tmpl
  - thes.tmpl
  - wsketch.tmpl
  - wsdiff.tmpl
- User testing:
  - testing_questionnaire.pdf – questionnaires for users
  - testing_feedback.csv – full feedback form
- Text of this thesis

ANNEX II

The credentials for access to the local installation:

Full URL:	http://corpora.fi.muni.cz/xkocinc/
Username:	demo
Password:	password


أسم
دولة
صورة
رأي
أمر
موقف
مصلحة
قوة
حق
نظام
عمل
ثقافة
شخصية
دور
خدمة
علاقة
حياة
فكر
قيادة
إنسان
نفس
مجتمع
تجربة
حركة
علم
تاريخ
شعب
عالم
بلد
حرية
تعليم


commitment
acceptance
confidence
freedom
friendship
independence
understanding
honour
pleasure