Computational Linguistics in the Netherlands Journal 10 (2020) 147–160 Submitted 10/2020; Published 12/2020

SnelSLiM, a user-friendly and fast tool to perform better keyword analysis through Stable Lexical Marker Analysis

Bert Van de Poel[*] BERT.VANDEPOEL@UCLOUVAIN.BE
Dirk Speelman[**] DIRK.SPEELMAN@KULEUVEN.BE

[*] *Centre for English Corpus Linguistics, Institute for Language & Communication, UCLouvain, Belgium*

[**] *Quantitative Lexicology and Variational Linguistics, KU Leuven, Belgium*

Abstract

This paper introduces snelSLiM, a corpus linguistics tool to perform keyword analysis using Stable Lexical Marker Analysis. SLMA goes beyond regular keyword analysis by assuring stability across the documents within a corpus and can be used with a contrastive or specific reference corpus as well as a general reference corpus. With snelSLiM, it is easy to perform SLMA as well as explore the resulting keywords through a user-friendly web interface. Great performance, support for many corpus formats, collocational analysis of keywords and visualizations make it an attractive tool for initial exploration of new corpora. A demo version is available on http://demo.snelslim.org/

1. Introduction

There is a wide array of software tools to make corpus analysis more available to those who lack the skills or interest to implement common linguistic analyses in e.g. Python or R. Keyword analysis usually offers a straightforward way to explore a new corpus and get a quick idea of what is potentially interesting to investigate further. While many common corpus linguistics tools have support for some form of keyword analysis, it is usually not the main feature of the software and therefore has certain disadvantages when it comes to methodology and/or user-friendliness. SnelSLiM[1] is an open source corpus linguistics tool with keyword analysis as its main functionality. This way, user-friendliness, methodology and performance are fine-tuned for the specific task, but of course this limits the tool's general use.

2. Method

Most commonly, keyword analysis is performed by condensing a specialised target corpus and a general reference corpus to frequency lists of all or the most frequent lowercased tokens. It is common practice to remove case variations so all occurrences of a token are added together, whether sentence-initial or not. Based on the frequency lists, standard contingency tables are built for each token to perform a keyword measure on. Different tools use different measures, ranging from log likelihood and chi squared statistical tests, to methods such as simple maths and log ratio which focus on the ratio of relative frequency. While using different keyword measures, all tools share the use of global frequency lists for each corpus. Though common, this method has the tendency of losing relevant insights on a per document level, by condensing frequencies. To prevent this loss of information, snelSliM uses Stable Lexical Marker Analysis.

Stable Lexical Marker Analysis was first introduced in (Speelman et al. 2006) and (Speelman et al. 2008) and then further extended in (De Hertog et al. 2014). Based on concepts from (Scott 1997) and (Dunning 1993), keyword analysis is performed based on per document frequency lists,

[1] Earlier versions of the snelSLiM software were developed as part of my bachelor paper, master thesis and advanced master thesis at KU Leuven under the supervision of the second author.

cross-referencing each document of the target and reference corpus. This method ensures keyness of keywords across documents, making them stable for the target corpus based on the reference. This prevents specific documents with high token frequency from over-influencing the global result for the entire corpus. It also loosens the requirements for the reference to be rather general, giving users of the method the option to use a contrastive specialised corpus as a reference that pairs well with the target.

Stable Lexical Marker Analysis relies on the log likelihood statistical test for each document combination's contingency tables to assess statistical significance, and then measures the effect size based on the log odds ratio. The log likelihood value is compared to a cut-off value. If the value is higher than the cut-off, the word is a keyword or marker for this document combination. The cut-off value is based on the quantile function of the chi-squared distribution with a degree of freedom of one and a probability based on the user's preference (99.9% by default). If the result is larger than the cut-off value and therefore statistically significant, the token is a keyword for either the target or the reference corpus. To decide, the frequency relative to the entire document is compared between the selected document from the target and the selected document from the reference. If the keyword in the target document has a higher relative frequency, the keyword is attracted to the target corpus for that document combination, otherwise it is repulsed. For every token, the attracted and repulsed results are counted. Based on which one is higher, the token is considered attracted or repulsed. An absolute score is calculated based on subtracting the number of repulsed significant combinations from the attracted significant combinations. A normalized score is also calculated based on the proportion of the absolute score and the total amount of comparisons, meaning a value of one shows the token was significant and attracted for each combination, and a value of minus one shows the token was significant and repulsed for each combination.

To calculate the effect size, the same contingency table mentioned earlier is used as the basis for a log odds ratio score (the log of the division of the relative frequencies in both corpora) for each significant combination. The average is used for the effect size score, while the minimum, maximum and standard deviation are also calculated for further contextualization of the average. The effect size makes it possible to sort the results from most to least impactful for the target corpus. While many methods are open to the use of different association measures, such as is the case with regular keyword analysis, SLMA always implies the use of log likelihood as the statistical test (with the cut-off based on the quantile function of the chi-squared distribution) and log odds ratio as the effect size measure. SnelSLiM therefore strictly adheres to the well-established SLMA methodology as outlined in (De Hertog et al. 2014).

A methodologically related concept is that of multi-word keywords, sometimes referred to as terms or n-gram keywords. While some tools such as Sketch Engine and #LancsBox have support for this kind of keyword analysis, it is not currently a feature of snelSLiM. Stable Lexical Marker Analysis does have methodology in place for multi-word keyword extraction (De Hertog et al. 2012), which may be added into snelSLiM in the future.

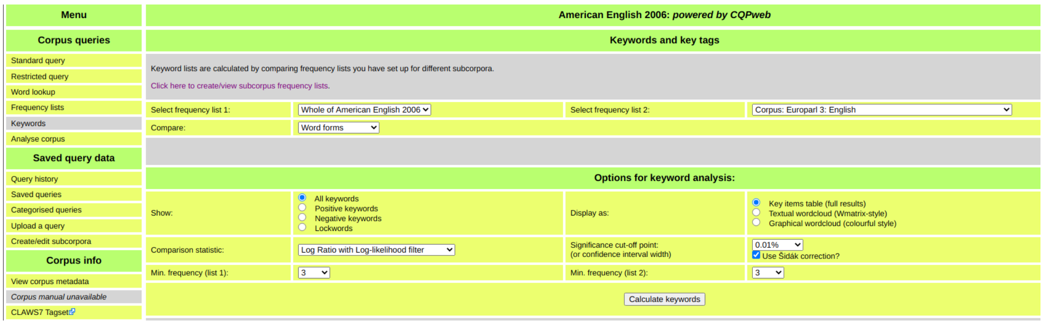
3. Comparison to existing tools

While snelSliM opts for a more rigorous methodology for its extraction of keywords, there are also other aspects of the software that are different compared to other corpus tools. For this article, a comparison will be made to AntConc (Anthony 2019), Sketch Engine (Kilgarriff et al. 2004), #LancsBox (Brezina et al. 2020), CQPweb (Hardie 2012) and BNCweb (Hoffmann et al. 2008). All the tools snelSLiM was compared to use standard keyword analysis with global frequency lists. The most commonly used implementation of Stable Lexical Marker Analysis is the R library mclm[2] written by the second author (Speelman 2021). While mclm offers SLMA as one of its core feature, it is not a stand-alone piece of software (compared to all other software in this article) and requires

[2] https://wwwling.arts.kuleuven.be/mclm/

the knowledge to prepare a corpus and load it in R, then perform the analysis, and finally extract the results. Due to its programming in R instead of Rcpp (Seamless R and C++ Integration), it also lacks in performance, making it slow for larger corpora with many documents and requiring more memory than available on a standard computer for a million word corpus. It is therefore not considered in this comparison as it serves another use and userbase.

Tools like AntConc and #LancsBox are run locally instead of on a remote server accessible through a web browser. This makes it easy to input different corpora into the tool. They however have limited support for the many XML formats that are popular in corpus linguistics. Web tools such as CQPweb and BNCweb are fairly limited when it comes to the use of different corpora. Obviously BNCweb only exposes the BNC corpus, while installations of CQPweb contain the corpora installed by the administrator of that instance. Both snelSLiM and Sketch Engine try to combine the ease of use of an external web tool (which can have more computational power available and makes access from different devices easier) with the option to supply new corpora. Sketch Engine contains its own tools to crawl the web to compile corpora, as well as an option to upload files. When uploading XML formats, Sketch Engine has tools available to decode the XML annotation. SnelSLiM comes with support for a large amount of popular corpus formats: Alpino XML, TEI XML BNC/Brown Corpus Variant, CoNLL(-U), DCOI XML, Eindhoven corpus, FoLiA XML, Gysseling corpus, NLPL OPUS, PRAAT TextGrid, XCES GrAF, plain text (with or without metadata tags that require filtering, such as a tag for a pause, a gesture, or text from a moderator or instructor) and generic XML with a user-supplied XPath query. This means that snelSLiM and Sketch Engine have very similar support for existing corpora, but with snelSLiM offering ready-made format parsers for users with less knowledge about corpus formats and annotation. Sketch Engine does have a unique feature in its corpus compilation functionality through web crawling.



American English 2006: *powered by CQPweb*

Keywords and key tags

Keyword lists are calculated by comparing frequency lists you have set up for different subcorpora.

Click here to create/view subcorpus frequency lists.

Select frequency list 1: Whole of American English 2006

Select frequency list 2: Corpus: Europarl 3: English

Compare: Word forms

Options for keyword analysis:

Show: ● All keywords ○ Positive keywords ○ Negative keywords ○ Lockwords

Display as: ● Key items table (full results) ○ Textual wordcloud (Wmatrix-style) ○ Graphical wordcloud (colourful style)

Comparison statistic: Log Ratio with Log-likelihood filter

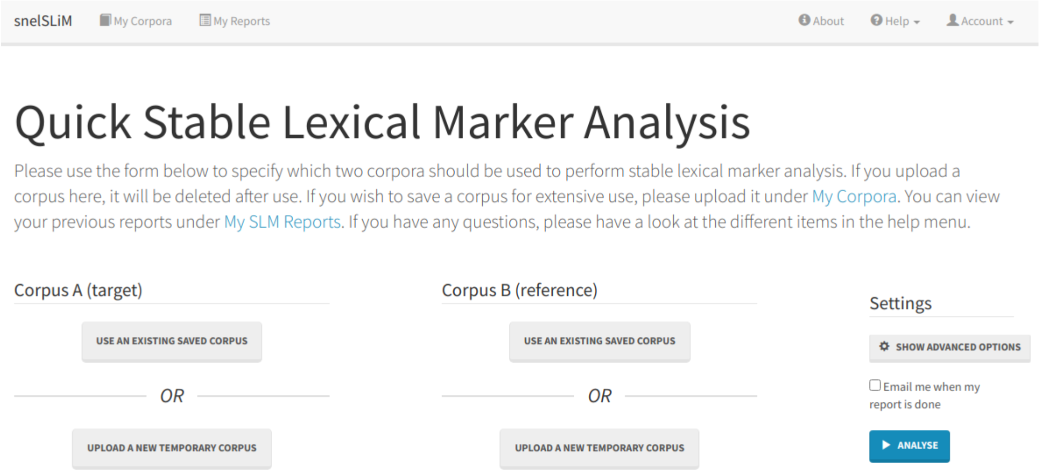
Significance cut-off point: (or confidence interval width) 0.01% ☑ Use Šidák correction?

Min. frequency (list 1): 3

Min. frequency (list 2): 3

Figure 1: Keywords interface of CQPweb

For many of the existing tools, keyword analysis is not a main feature. Usually, the user is expected to already have an interest in certain tokens within the corpus. This results in a sensible focus on lookup queries, concordance and collocation/word sketches. For this reason, keyword analysis within all tools except Sketch Engine is somewhat daunting for a user with limited prior knowledge about the methodology. In case of AntConc, a reference corpus has to be loaded through the tool preferences, where a user can also control the keyword measure and effect size measure. While not too difficult in and by itself, it can prove overwhelming or confusing for some users. Similarly, #LancsBox gives a fairly good overview of top keywords once a user has managed to perform the keyword analysis. It is hard to imagine however that a user will be able to figure out without the user guide that keyword analysis can be performed by opening up the minimized bottom panel on the words tab and then dragging the representations of both corpora on top of each other. Keyword Analysis on BNCweb and CQPweb results in large forms where the target and reference can be selected, touching upon the earlier limitation in supplying custom corpora, as well as several technical settings for the analysis such as the keyword measure, minimum frequencies and significance cut-off. Sketch Engine is fairly successful in simplifying the process, splitting up keyword analysis into a basic and a more advanced option. While the basic keyword analysis presumes defaults and starts the analysis, a more experienced user can control target subcorpus selection, the reference corpus, the N value for simple maths and several other settings. SnelSLiM follows a similar approach to Sketch Engine when it comes to user-friendliness. By default, the user is simply shown a form to select or upload a target and reference corpus with a checkbox to receive an email once the analysis has finished. Under the advanced options, a more experienced user can control how many items are analysed, the statistical significance cut-off, disable visualizations and control the settings for collocational analysis.



Quick Stable Lexical Marker Analysis

Please use the form below to specify which two corpora should be used to perform stable lexical marker analysis. If you upload a corpus here, it will be deleted after use. If you wish to save a corpus for extensive use, please upload it under My Corpora. You can view your previous reports under My SLM Reports. If you have any questions, please have a look at the different items in the help menu.

snelSLiM
My Corpora
My Reports
About
Help
Account
Corpus A (target)
USE AN EXISTING SAVED CORPUS
OR
UPLOAD A NEW TEMPORARY CORPUS
Corpus B (reference)
USE AN EXISTING SAVED CORPUS
OR
UPLOAD A NEW TEMPORARY CORPUS
Settings
SHOW ADVANCED OPTIONS
□ Email me when my report is done
ANALYSE

Figure 2: Main view of snelSLiM without advanced options opened

4. Unique Features

Beyond the obvious fact that most corpus linguistics tools focus on offering a wide range of analyses, and snelSLiM focusses on doing one specific task as well as possible, there are many unique features that may make it an attractive piece of software for a wide range of potential users.

As mentioned before, snelSLiM is accessible through an attractive web interface that will quickly feel familiar to most users. While the intricacies of the analysis can still be controlled by a user, they are hidden by default. Even though many web tools are constrained by the lack of performance in programming languages such as PHP, snelSLiM tries to pick the programming languages it uses carefully to balance ease of use and performance. Under the hood, snelSLiM harnesses the power of binaries compiled from code in Go combined with xmllint and common unix shell tools, to swiftly extract tokens from corpora, generate frequency lists and perform multithreaded Stable Lexical Marker Analysis. The web interface itself is written in PHP and calls upon the Go binaries to perform the heavy lifting. For visualizations and interactivity, some JavaScript and Vega is used. Bootstrap (a popular, theme-able library for creating interfaces, conceived by Twitter) gives the interface a familiar yet unique look. By building all of the interface of snelSLiM on top of PHP, it is even possible to run snelSLiM on standard shared hosting[3] as long as they do not prevent the execution of binaries from PHP. In cases where corpora contain hundreds of documents and millions of words, it is advised to use a Virtual Private Server (VPS) or physical server with some more

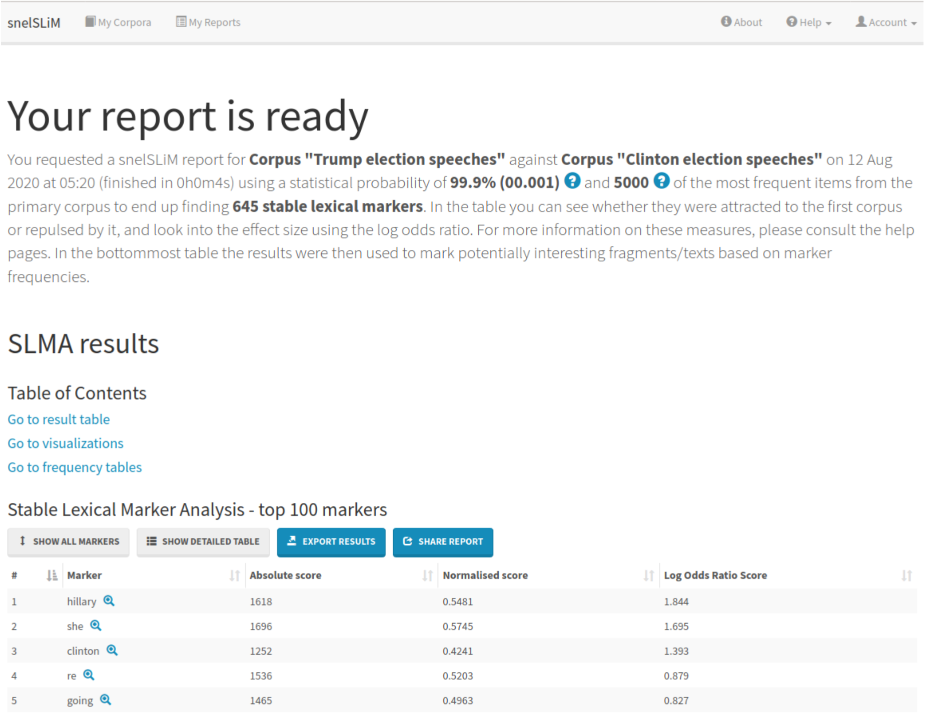
[3] Standard web hosting offered by many companies for simple websites, requiring no knowledge of server maintenance but limited in functionality (usually offering PHP with a MySQL database). The server is shared among a large amount of customers, all using a small portion of the infrastructure.

power and storage, as analyses may take a long time and use a larger amount of resources. While larger analyses will use a lot of CPU power to perform all calculations, the memory footprint of the Go binaries remains quite limited. Meaning that in most cases, the web server software itself will use more memory than a running snelSLiM analysis.

With snelSLiM running on a remote device, it is very convenient for users who want to perform an analysis with corpora that contain many documents and therefore might take more than a few seconds. While the page will automatically refresh periodically to check whether a report has been generated (or an error encountered), users are free to close their browser or do something else on their computer without fearing their analysis might stop when their session expires. There is also an easy checkbox so users can receive notification by email when their report is available.

SnelSLiM contains simple interfaces to manage corpora and previous reports, has detailed help pages on the many supported corpus formats and the used statistics, as well as a PDF manual with screenshots, and easy to manage accounts and permissions. The report of an analysis itself goes much beyond simply listing the keywords and their scores. The report starts with a simple overview which corpora were compared, with which statistical probability, how many tokens were investigated and how many of those were successful keywords. This is followed by the standard table of keywords that most software includes, with the option to look more deeply into a specific keyword by pressing an icon. This displays the frequency of the token within the target corpus, as well as its relative position in a global frequency list and what percentage of the corpus is taken by that token. After the specific scores for the Stable Lexical Marker Analysis for that token, a list is included with the frequency of the token within each document of the target corpus, as well as (optionally) its collocates. Collocational analysis is performed using log Dice based on the window size set by the user. This same association measure is used by Sketch Engine for its word sketches. Similarly, Sketch Engine makes it easy to look at the collocates of a keyword, while other tools link to the concordances. Further in the main report, a treemap visualization shows the distribution of keywords across the target corpus. In this visualization, each rectangle represents a file in the target corpus. The size of the rectangle is by default based on the number of keywords in that file, while the colour is based on whether the majority of a file's keywords are attracted or repulsed. If the corpus is consistent in size and make-up, this visualization will not yield very interesting results, displaying only blue rectangles of fairly uniform size. If however some documents stand out from the rest of the corpus for some reason (size or colour), they may contain far fewer keywords of the target corpus (or few tokens in general), or may mostly contain keywords that are typical of the reference corpus. This may help a user decide which files to investigate or may hint that more cleaning is required, including better tokenization and lemmatization or the need to filter out certain less representative files. In general, it is important that a user always investigates further if some rectangles are much larger or smaller than the others or if any are coloured red. Finally, for both the target and the reference, a table with keyword frequencies per document is included. Clicking on the documents in the treemap visualization or in the table for the target corpus, opens up a page with some of the raw frequencies and percentages of the file. Some space is reserved on this page for more information or visuals that may be introduced in the future.

Moving from a snelSLiM analysis to a basic publication or report is simplified by functionality to export the main results table to many different formats. Beyond a TSV or CSV for analysis in R or Python, it is also possible to export results to spreadsheeting software such as Microsoft Excel or LibreOffice Calc, as well as LaTeX tables or HTML, ready to copy to a word processor. The user is in control which columns are included. This way, it is much easier for users to include their results as an appendix to a publication or to share results among colleagues for further analysis in other tools.



Your report is ready

You requested a snelSLiM report for **Corpus "Trump election speeches"** against **Corpus "Clinton election speeches"** on 12 Aug 2020 at 05:20 (finished in 0h0m4s) using a statistical probability of **99.9% (00.001)** and **5000** of the most frequent items from the primary corpus to end up finding **645 stable lexical markers**. In the table you can see whether they were attracted to the first corpus or repulsed by it, and look into the effect size using the log odds ratio. For more information on these measures, please consult the help pages. In the bottommost table the results were then used to mark potentially interesting fragments/texts based on marker frequencies.

SLMA results

Table of Contents

Go to result table

Go to visualizations

Go to frequency tables

Stable Lexical Marker Analysis - top 100 markers

#	Marker	Absolute score	Normalised score	Log Odds Ratio Score
1	hillary	1618	0.5481	1.844
2	she	1696	0.5745	1.695
3	clinton	1252	0.4241	1.393
4	re	1536	0.5203	0.879
5	going	1465	0.4963	0.827

Figure 3: The beginning of a snelSLiM report

5. Case Studies

To make these claims about methodology, user-friendliness and unique features more tangible, the following three case studies will show the use of snelSLiM to perform keyword analysis and gain enough insight to start research.

5.1 British National Corpus (1994): Baby

In this first very basic case study, two subcorpora of the BNC Baby are used as target and reference, specifically conversations (dem) and newspapers (news). The BNC uses a variant on the TEI XML standard, which is automatically detected by snelSLiM. This is a simplified and limited first example to illustrate the method and show the difference between spoken and written English as well as informal and formal. The top results (by effect size) are typical discourse markers used in spoken English (and some like *yeah* more informal settings). Beyond that, first and second person singular personal pronouns would make very little sense in most newspaper articles. Spoken variants such as *cos* instead of *because* and *na* instead of *no* are also to be expected. Of course, no user would expect shocking results from a corpus as thoroughly researched as the BNC.

Marker	Absolute score	Normalised score	Log Odds Ratio Score
yeah	2596	0.892	4.378
oh	2390	0.821	3.669
mm	2015	0.692	3.053
you	2771	0.952	2.963
er	1879	0.646	2.737
i	2737	0.941	2.560
erm	1814	0.623	2.557
get	2469	0.848	2.266
cos	1525	0.524	2.130
na	1506	0.518	2.102

Table 1: Top results (based on effect size) for the snelSLiM analysis of the BNC Baby conversations (dem) with the BNC Baby newspapers (news) as a reference using default settings

In the treemap visualization of this report, we can see that while all files contain a majority of keywords that are attracted to the target corpus, there is a clear size difference. The smallest document in the conversations subcorpus is 1453 tokens, while the largest has 122,092. In this case it does not seem to pose much of an issue, as nearly 1500 tokens is still a respectable and representative text, but in cases where a document may just contain a few sentences, the results might become problematic. Therefore, this visualization paints a compelling picture of the composition of the target corpus.

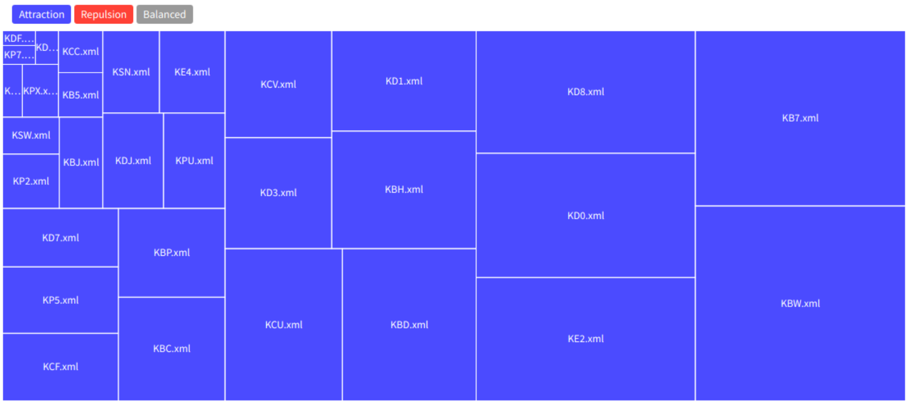


Figure 4: Treemap visualization in snelSLiM for a Stable Lexical Marker Analysis of the BNC dem against the BNC news corpus

5.2 Trump and Clinton Corpus

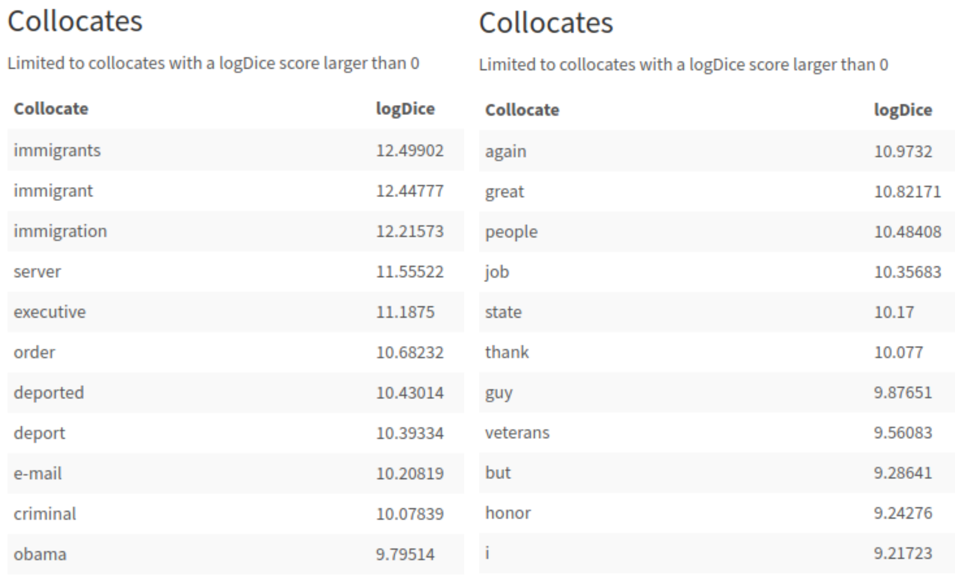
The next case study looks at the Trump and Clinton election corpus. It is a public domain corpus compiled by D. W. Brown of The Grammar Lab and contains all the election speeches from the 2016 US presidential election, starting from the respective democratic and republican nomination acceptance speeches. The corpus is stored in plain text with html tags for metadata such as questions from a moderator or applause. SnelSLiM has support for this kind of corpus format and can easily filter out the tags. The Trump and Clinton subcorpora are included in the snelSLiM demo available on http://demo.snelslim.org/. The two candidates were considered polar opposites by both sides of the US American political spectrum, and these two subcorpora should therefore be in great contrast to one another. In this case study, Trump is first used as the target and Clinton as the reference, and then reversed.

It is easy to already formulate some hypotheses about the results when using these two subcorpora interchangeably for analysis. While Hillary Clinton is a fairly standard politician who will talk about general political concepts such as the economy, American families, international politics, education and religion, Donald Trump is exceptional in his rhetoric. This means that some may expect keywords in line of *make America great again*, *Mexican rapists*, *build the wall*, *Mexico will pay*, *illegal immigrants*, *China*, etc. It is however important to keep an open mind. Stability plays an important role within SLMA, which may result in some popular but short lived keywords not ranking very high.

Since the two presidential candidates are of course of the opposite sex and often talk about each other, it makes sense that personal pronouns and names are high in the results for both analyses. Beyond that, many short and rather function-like words are used by Trump, perhaps echoing the journalists claiming his vocabulary is more limited, but perhaps also because of his erratic nature. It is important to keep in mind that stability across speeches gives clearer results in SLMA. In the top 10 results for Trump no stereotypes are present, but slightly further in the list, *great*, *win*, *jobs*, *china*, *obamacare*, *illegal*, *mexico* and *wall* show up. Surprisingly, *mexican* is not an attracted keyword but repulsed, meaning it is a keyword for the Clinton subcorpus (but with a very small effect size). Again this may point towards the inconsistency of Trump in using certain tropes.

Marker	Absolute score	Normalised score	Log Odds Ratio Score
hillary	1618	0.548	1.844
she	1696	0.575	1.695
clinton	1252	0.424	1.393
re	1536	0.520	0.879
going	1465	0.496	0.827
they	1665	0.564	0.807
very	912	0.309	0.748
her	406	0.138	0.410
will	579	0.196	0.267
t	525	0.178	0.256

Table 2: Top results (based on effect size) for the snelSLiM analysis of the Trump corpus with Clinton corpus as a reference using default settings



Collocates

Limited to collocates with a logDice score larger than 0

Collocate	logDice
immigrants	12.49902
immigrant	12.44777
immigration	12.21573
server	11.55522
executive	11.1875
order	10.68232
deported	10.43014
deport	10.39334
e-mail	10.20819
criminal	10.07839
obama	9.79514

Collocates

Limited to collocates with a logDice score larger than 0

Collocate	logDice
again	10.9732
great	10.82171
people	10.48408
job	10.35683
state	10.17
thank	10.077
guy	9.87651
veterans	9.56083
but	9.28641
honor	9.24276
i	9.21723

Figure 5: Screenshot of the top collocates for *illegal* and *great* in the snelSLiM report for the snelSLiM analysis of the Trump corpus with Clinton corpus as a reference using default settings

When looking at the collocates for *illegal*, the focus is clearly on immigration but also on the Clinton e-mail server and Obama. The collocates for *great* show how ubiquitous Trump's use of that word is. Beyond the word *again* and the word *great* itself, a whole list of other great things have very high Dice scores: *people*, *job*, *state*, *guy*, *veterans*, *honor*, *deals*, *wall*, etc.

Similarly to Trump, personal pronouns and names feature prominently at the top of the results for Clinton. However, *together*, *rights* and *economy*, very stereotypical political terminology, are already present in her top 10. Throughout the results, this trend continues with *nuclear*, *college*, *weapons*, *gun*, *tax*, *church*, *faith*, *middleclass*, etc. One of the more peculiar results is *Khan*. This refers to the deceased US Army Captain Humayun Khan, whose parents spoke during the Democratic National Convention. It may be an interesting insight that this name was mentioned consistently enough to be stable across documents as a keyword.

Marker	Absolute score	Normalised score	Log Odds Ratio Score
his	1041	0.353	0.964
he	1332	0.451	0.929
together	600	0.203	0.600
work	682	0.231	0.579
donald	564	0.191	0.567
election	447	0.151	0.493
america	547	0.185	0.479
families	398	0.135	0.462
rights	372	0.126	0.457
economy	323	0.109	0.364

Table 3: Top results (based on effect size) for the snelSLiM analysis of the Clinton corpus with Trump corpus as a reference using default settings

While length is mostly consistent across both candidates, the stability of keywords across documents is less the case for Trump. Clinton is consistent in both situations. As shown below, a few of Trump's speeches contain more keywords attracted to Clinton's speeches than his own. This could potentially be easily explained by an unremarkable practicality, but may also expose some interesting aspects about those specific speeches. Therefore, this case study does not only illustrate what to expect when comparing two clearly contrasting corpora, but also shows what to look for in the treemap visualization for further investigation along these lines.

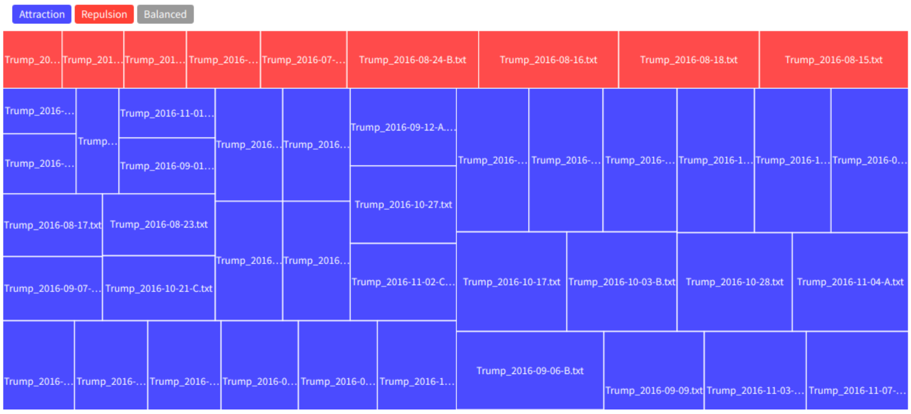
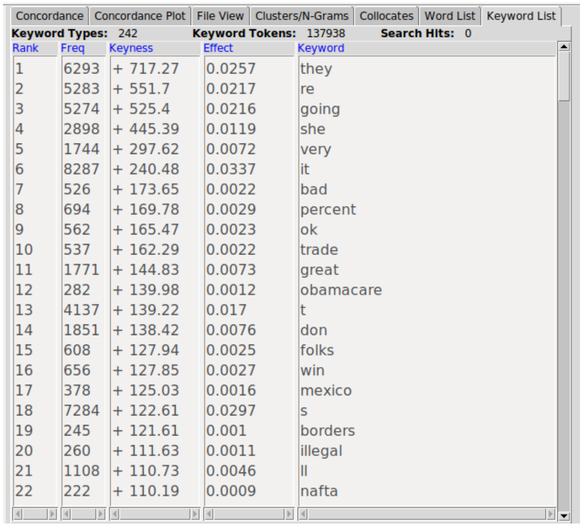


Figure 6: Top part of the treemap visualization in snelSLiM for a Stable Lexical Marker Analysis of the Trump election subcorpus against the Clinton election subcorpus

As the Trump and Clinton subcorpora are both stored in plain text, they are compatible with several of the tools snelSLiM was compared to. When performing keyword analysis in AntConc, #LancsBox and SketchEngine with default settings, it is not only clear that different methodology yields very different results, but that changes in parameters also have a noticeable impact.

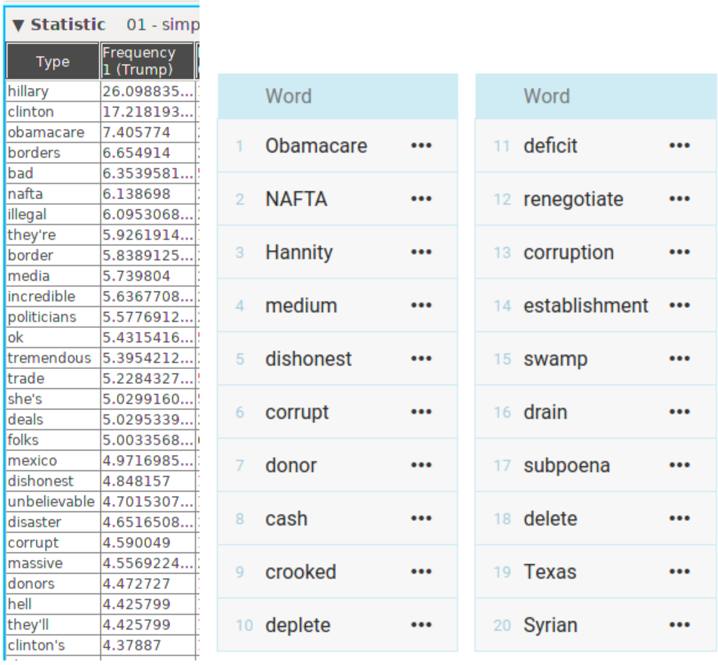


Keyword Types: 242 **Keyword Tokens:** 137938 **Search Hits:** 0

Rank	Freq	Keyness	Effect	Keyword
1	6293	+ 717.27	0.0257	they
2	5283	+ 551.7	0.0217	re
3	5274	+ 525.4	0.0216	going
4	2898	+ 445.39	0.0119	she
5	1744	+ 297.62	0.0072	very
6	8287	+ 240.48	0.0337	it
7	526	+ 173.65	0.0022	bad
8	694	+ 169.78	0.0029	percent
9	562	+ 165.47	0.0023	ok
10	537	+ 162.29	0.0022	trade
11	1771	+ 144.83	0.0073	great
12	282	+ 139.98	0.0012	obamacare
13	4137	+ 139.22	0.017	t
14	1851	+ 138.42	0.0076	don
15	608	+ 127.94	0.0025	folks
16	656	+ 127.85	0.0027	win
17	378	+ 125.03	0.0016	mexico
18	7284	+ 122.61	0.0297	s
19	245	+ 121.61	0.001	borders
20	260	+ 111.63	0.0011	illegal
21	1108	+ 110.73	0.0046	ll
22	222	+ 110.19	0.0009	nafta

Figure 7: Keyword analysis using AntConc of the Trump election subcorpus against the Clinton election subcorpus

By default, AntConc uses log likelihood for its statistical test and the dice coefficient for effect size, while both #LancsBox and SketchEngine use the simple maths method for keyword analysis.



▼ Statistic 01 - simp

Type	Frequency 1 (Trump)
hillary	26.098835...
clinton	17.218193...
obamacare	7.405774
borders	6.654914
bad	6.3539581...
nafta	6.138698
illegal	6.0953068...
they're	5.9261914...
border	5.8389125...
media	5.739804
incredible	5.6367708...
politicians	5.5776912...
ok	5.4315416...
tremendous	5.3954212...
trade	5.2284327...
she's	5.0299160...
deals	5.0295339...
folks	5.0033568...
mexico	4.9716985...
dishonest	4.848157
unbelievable	4.7015307...
disaster	4.6516508...
corrupt	4.590049
massive	4.5569224...
donors	4.472727
hell	4.425799
they'll	4.425799
clinton's	4.37887

	Word
1	Obamacare
2	NAFTA
3	Hannity
4	medium
5	dishonest
6	corrupt
7	donor
8	cash
9	crooked
10	deplete

	Word
11	deficit
12	renegotiate
13	corruption
14	establishment
15	swamp
16	drain
17	subpoena
18	delete
19	Texas
20	Syrian

Figure 8: Keyword analysis using #LancsBox (left) and SketchEngine (right) of the Trump election subcorpus against the Clinton election subcorpus

It is clear that while there is some overlap between the top results of snelSLiM, AntConc, #LancsBox and SketchEngine, large differences are also quite obvious. It is also remarkable that while #LancsBox and SketchEngine share the same method, they use different tokenization, different filters and different default settings for the keyword analysis itself. This results in some very different keywords. As there is currently no gold standard for measuring the success of keyword analysis, it is open for debate which of these applications yields the best results. The preference for SLMA within snelSLiM is based on the concept of stability between the different documents within a corpus. This stability becomes clear when looking at keywords such as *nafta*. This keyword is very highly ranked within all other tools, but is only the 180th keyword in the snelSLiM report. Its absolute score is 6, meaning that of the 2952 text comparisons that were made (the Trump subcorpus contains 82 documents, the Clinton subcorpus 36), only a marginal amount had statistical significance. We see this reflected in the very low effect size of 0.007. This concept is something a user should keep in kind when considering keyword methodology. While this alone should not be a reason to discard traditional keyword analysis, it should serve as a warning on how to interpret keywords depending on which tool is used.

5.3 Europarl Corpus

This final case study uses the English Europarl-8 corpus from the NLPL OPUS project. As part of the OPUS project, they are in the OPUS XML format, which can be parsed by snelSLiM. This corpus features English translations of the debates in the European parliament. The subcorpora from the years 1996 up to 2001 are included in the snelSLiM demo available on http://demo.snelslim.org/. This case study specifically compares 2001 as target with 1996 as reference.

When comparing a corpus like Europarl to a general reference, all kinds of political and bureaucratic terminology characteristic of the European parliament would show up. If a user is interested in topics of specific years or changes within the manner of communication within the European parliament, using a specific year or collection of years as a target and reference may prove more fruitful.

2001 was a tumultuous year in European and world history. This means that many interesting topics were on the European agenda, and this is clear from the results. The 9/11 terrorist attacks and following war in Afghanistan were high on the agenda based on the analysis. But the Nice treaty and preparations for its follow-up in the Lisbon treaty were also clearly on the agenda to reform the European structure. For the European institutions, the foot-and-mouth disease outbreak in the UK was of course a major tragedy closer to home. But as the analysis makes clear, the Belgian EU presidency, pension schemes, liberalisation, Macedonia, and many other political topics were also on the agenda at the time. While the case study of Trump and Clinton had quite a lot of noise in the results, most keywords for this analysis clearly correspond with political activity and discussion at the time, yielding the purest result of these three use cases. Of course, the size of the Europarl corpus (2.3 million tokens for 1996 and 3.7 million tokens for 2001) plays an important role. Finally, it is surprising to see *s* as the first keyword and with such a high effect size. This is caused by an inconsistency in the corpus where *'s* was sometimes tokenized in 2001 as two tokens, while it was almost always tokenized as a single token in 1996. This causes an interesting effect, so users should always remain critical before using results in publication.

Marker	Absolute score	Normalised score	Log Odds Ratio Score
s	2837	0.995	4.336
organisation	1579	0.554	1.424
nice	871	0.305	1.106
terrorism	743	0.261	0.959
terrorist	634	0.222	0.781
recognise	780	0.273	0.745
organise	625	0.219	0.677
candidate	615	0.216	0.658
afghanistan	475	0.167	0.634
safety	612	0.215	0.571
pension	440	0.154	0.564
reform	650	0.228	0.530
belgian	470	0.165	0.519
foot-and-mouth	383	0.134	0.503
directive	520	0.182	0.502
food	468	0.164	0.495
september	563	0.197	0.494
liberalisation	412	0.144	0.485
enlargement	431	0.151	0.483
lisbon	377	0.132	0.479
macedonia	356	0.125	0.462

Table 4: Top results (based on effect size) for the snelSLiM analysis of the Europarl 2001 corpus with Europarl 1996 corpus as a reference using default settings

6. Conclusion and Future Work

By mainly focusing on extracting keywords using Stable Lexical Marker Analysis, it was possible to optimize snelSLiM's performance and user-friendliness specifically for this task. SnelSLiM's wide array of features sets it apart from other tools for keyword analysis. This makes snelSLiM a good choice for those users who wish to investigate newly compiled or acquired corpora before diving in in detail. It takes some of the risk away when unsure what information might be available in the corpus, while requiring little wait time and only minor knowledge of the files of the corpus. Using visualizations, the user can easily decide whether certain documents in the corpus or corpus compilation steps need extra attention. Unexpected keywords can furthermore be explained using the available collocates and the distribution within the corpus documents. This makes snelSLiM a great tool for those eager to explore.

As an open source project, users can easily reach out with questions or feature suggestions on GitHub or even contribute code themselves. This means that the future of snelSLiM is open ended. More visualizations and further user experience enhancements have recently been added to snelSLiM, and its development will continue as long as there is demand for new features. There is also great potential for snelSLiM to become a popular tool for teaching corpus linguistics and for the first steps of research. It is clear that snelSLiM fills a void within corpus linguistics for an easy to use, open source and fast tool. Hopefully it can be a vector for both interesting research and for more tools to be developed by others.

References

Anthony, Laurence (2019), AntConc (version 3.5.8) [computer software]. Available from https://www.laurenceanthony.net/software.

Bertels, Ann, Dirk De Hertog, and Kris Heylen (2012), Etude sémantique des mots-clés et des marqueurs lexicaux stables dans un corpus technique (semantic analysis of keywords and stable lexical markers in a technical corpus) [in French], *Proceedings of the Joint Conference JEP-TALN-RECITAL 2012, volume 2: TALN*, ATALA/AFCP, pp. 239–252.

BNC Consortium (2005), BNC Baby version 2, *Distributed by Oxford University Computing Services on behalf of the BNC Consortium.* http://www.natcorp.ox.ac.uk/.

Brezina, Vaclav, Pierre Weill-Tessier, and Anthony McEnery (2020), #LancsBox v. 5.x [software]. Available at: http://corpora.lancs.ac.uk/lancsbox.

Brown, David W. (2017), Clinton-Trump corpus. Retrieved from http://www.thegrammarlab.com.

De Hertog, Dirk, Kris Heylen, and Dirk Speelman (2012), The prevalence of multiword term candidates in a legal corpus, *Proceedings of the 10th Terminology and Knowledge Engineering Conference (TKE2012): New frontiers in the constructive symbiosis of terminology and knowledge engineering*, Universidad Politecnica de Madrid, pp. 283–290.

De Hertog, Dirk, Kris Heylen, and Dirk Speelman (2014), Stable lexical marker analysis: a corpus-based identification of lexical variation, *in* Da Silva, Augusto Soares, editor, *Pluricentricity: Language variation and sociocognitive dimensions*, Vol. 24, Walter de Gruyter, pp. 127–141.

De Hertog, Dirk, Kris Heylen, Dirk Speelman, and Hendrik Kockaert (2010), A variational linguistics approach to term extraction, *Proceedings TKE 2010: presenting terminology and knowledge engineering resources online: models and challenges (on cd-rom)* pp. 229–248, Dublin city university.

Dunning, Ted E (1993), Accurate methods for the statistics of surprise and coincidence, *Computational linguistics* **19** (1), pp. 61–74.

Hardie, Andrew (2012), CQPweb - combining power, flexibility and usability in a corpus analysis tool, *International journal of corpus linguistics* **17** (3), pp. 380–409, John Benjamins.

Hoffmann, Sebastian, Stefan Evert, Nicholas Smith, David Lee, Ylva Berglund-Prytz, et al. (2008), *Corpus linguistics with BNCweb - a practical guide*, Vol. 6, Peter Lang.

Kilgarriff, Adam, Pavel Rychlý, Pavel Smrž, and David Tugwell (2004), The Sketch Engine (Itri-04-08), *Information Technology.*

Kilgarriff, Adam, Vít Baisa, Jan Bušta, Miloš Jakubíček, Vojtěch Kovář, Jan Michelfeit, Pavel Rychlý, and Vít Suchomel (2014), The Sketch Engine: ten years on, *Lexicography* pp. 7–36, Springer.

Koehn, Philipp (2005), Europarl: A parallel corpus for statistical machine translation, *MT summit*, Vol. 5, Citeseer, pp. 79–86.

Lexical Computing Ltd. (2015), Statistics used in Sketch Engine. https://www.sketchengine.eu/wp-content/uploads/ske-statistics.pdf.

Ruette, Tom and Freek Van de Velde (2013), Moroccorp: tien miljoen woorden uit twee Marokkaans-Nederlandse chatkanalen, *Lexikos* **23**, pp. 456–475.

Scott, Mike (1997), PC analysis of key words - and key key words, *System* **25** (2), pp. 233–245, Elsevier.

Speelman, Dirk (2021), *Mastering Corpus Linguistics Methods: A Practical Introduction with AntConc and R*, John Wiley & Sons Canada, Limited.

Speelman, Dirk, Stefan Grondelaers, and Dirk Geeraerts (2006), A profile-based calculation of region and register variation: the synchronic and diachronic status of the two main national varieties of Dutch, *Corpus linguistics around the world*, Brill Rodopi, pp. 181–194.

Speelman, Dirk, Stefan Grondelaers, and Dirk Geeraerts (2008), Variation in the choice of adjectives in the two main national varieties of Dutch, *in* Kristiansen, Gitte and René Dirven, editors, *Cognitive Sociolinguistics: Language Variation, Cultural Models, Social Systems*, Vol. 39, Walter de Gruyter, pp. 205–233.

Tiedemann, Jörg (2012), Parallel data, tools and interfaces in OPUS., *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC 2012)*, Vol. 2012, pp. 2214–2218.