

Word Rain as a Service: Making semantically structured word clouds available to everyone

Magnus Ahlthorp

Nakajima Koen Research Institute
Stockholm, Sweden
magnus@nakajimakoen.org

Maria Skeppstedt

Centre for Digital Humanities
and Social Sciences Uppsala
Department of ALM
Uppsala University, Sweden
maria.skeppstedt@abm.uu.se

Abstract

The Word Rain text visualisation technique is a novel approach to the classic word cloud that uses word embeddings to make the visualisation useful for exploring the word content of a text or corpus. Downloading and running code for generating word rain visualisations can, however, be prohibitively difficult or cumbersome for non-technical users and for casual evaluation. These use cases would consequently benefit greatly from a streamlined interface. We have therefore collected everything needed for generating word rain visualisations in a web-based service, and made it available as a SWELANG CLARIN K-centre resource. The web service, as well as the code for generating word rains, is made available as open source. The web service is deployed at: <https://wordrain.isof.se>.

1 Introduction

Word clouds are widely used for informally illustrating the most prominent words in a text or corpus. Their popularity likely stems from their usefulness as a graphic design element, the simple, self-explanatory nature of the visualisation, as well as from the prevalence of easy-to-use tools for generating word clouds. The visualisation consists of a graph, where the words are displayed in a font size that reflects the prominence of the words. Typically, the more frequent a word in the text visualised, the larger the font size used for displaying the word. The words are often positioned in a random or alphabetical order, in a fashion that aims to maximize the visual design of the word cloud.

Traditional word clouds are, however, also criticised because of the lack of semantic relevance in the placement of words (Barth et al., 2014). The reader can therefore be misled into thinking there is a relationship between nearby words where there is none. The absence of a semantically meaningful word positioning also makes the word cloud visualisation unsuitable as a tool for practically exploring and analysing texts (Cao & Cui, 2016; Hicke et al., 2022). Since the traditional word cloud does not provide any guidance to where in the graph to zoom in to read words displayed in a small font size, the inclusion of less prominent words in the graph has no practical text exploration purpose – only a cosmetic function. In addition, typical text analysis tasks, e.g., to manually create semantic categories among the most prominent words in a text, or to compare the content of pairs of texts, are also very difficult to carry out when the words are positioned in a non-semantically relevant order.

Despite the critique directed towards the traditional word clouds as a tool for practically exploring and analysing texts, the word clouds are often used for these purposes (Hicke et al., 2022). We hypothesise that the aforementioned prevalence of easy-to-use tools for generating traditional word clouds – and the absence of similar tools for generating more practically useful visualisations – is an important reason for word clouds still being used for these tasks. The project described here therefore aims to direct researchers – as well as people in general – to the novel and more practically useful Word Rain text visualisation technique. By streamlining the process of generating word rains to the point where it is as easy to generate as a word cloud is today, our goal is to simplify the creation of text visualisations that can be used practically for exploring and analysing texts.

We will here start by providing a background which describes the Word Rain text visualisation technique. Then, we will concretise the theoretical description by discussing example word rains generated using the Word Rain web service. Thereafter, we will provide a practical description for how to use the Word Rain web service to generate word rain(s) from one or several texts. We will then conclude with a summary of the work conducted and by outlining possible future directions for the Word Rain web service.

2 The Word Rain text visualisation technique

There are many tools for visualising prominent words in a text that build on animation and/or more or less advanced user interaction (Liu et al., 2015; Wang et al., 2018; Xie et al., 2024). While such functionality increases the possibilities to carry out different types of text exploration and text analysis tasks, it does not – as the word cloud – provide a simple, self-explanatory static graph that, for example, can be included in an article or printed on a poster. That is, these tools are not suitable replacements to the traditional word cloud.

Other approaches that extend the traditional word cloud by generating a static graph, and which can be used in the same contexts as the word cloud, typically focus on solving only one of the problems associated with the traditional version of the visualisation technique. There are, for instance, approaches that make it easier to compare pairs of texts, either by generating graphs that simultaneously visualise several texts or by generating series of graphs where a word is always given the same position in the graph (Burch et al., 2014; Cui et al., 2010; Diakopoulos et al., 2015; Herold et al., 2019; Lee et al., 2010). There are also other approaches where a visualisation similar to the traditional word clouds is generated, except that semantically similar words are positioned close to each other (often in combination with a semantically motivated colour coding), and that aim to solve the problem of word clouds not supporting a manual categorisation of prominent words (Barth et al., 2014; Wu et al., 2011; Xu et al., 2016).

With the Word Rain technique (Skeppstedt et al., 2024), in contrast, we propose a solution to all of the above-mentioned problems associated with traditional word clouds – a solution that still produces a static, self-explanatory graph as its output. The most important difference from the traditional word cloud is that the Word Rain technique uses the position on the x-axis, as well as the position on the y-axis for conveying information. The semantic relevance of the word positioning is achieved by letting a word’s position on the x-axis represent the semantics of the word. The position is automatically determined by using multi-dimensional word embedding vectors that represent the words and reducing the vectors to one dimension. The position of a word on the y-axis is, instead, primarily based on word prominence. To avoid words overlapping when they have a similar position on the x-axis, less prominent words have to yield to more prominent ones. That is, the less prominent word “rains down” in the graph, until it reaches a lower y-position where its extension no longer overlaps with the extension of a more prominent word.

With this technique, words with a similar meaning will be positioned close to each other on the x-axis, creating (partly) vertical clusters of semantically similar words, with the more prominent ones typically at the top of the cluster. Thereby, the horizontal position of the words, and the clusters of similar words, can assist the user when manually creating categories of prominent words in a text. The semantic word positioning also has the effect that the more prominent words – which are displayed in a large font size – can guide the user to semantically interesting areas in the graph, where the user can zoom in and read semantically similar words displayed in a small font size. Finally, several word rains can be generated with the same embedding projection on the x-axis, making it possible to compare corpora. This comparison can be conducted on the level of each individual word, but semantic clusters of words can also be compared.

The word embedding model used for creating the semantic word positioning could either consist of a pre-trained word embedding model, or the user could train their own model, e.g., on the texts that are to be visualised. The multidimensional word embedding vectors are projected down to the one-dimensional x-axis using t-SNE dimensionality reduction.

There are two configurations for determining word prominence, one is *term frequency*, i.e. the frequency of the words in the text visualised, and the other is *term frequency–inverse document frequency*,

i.e., a measure that down-weights words that occur in many texts. The word prominence is not only indicated by a word’s position on the y-axis, but – similar to the traditional word cloud – its prominence is also used to determine the font size used for displaying the word. In addition, there is a row of bars associated with the words in the graph, where the height of a bar is proportional to the prominence of its associated word. The bars do not only function as an additional indication of word prominence, but also emphasise which semantic regions on the x-axis that are heavily populated in the text.

We have previously showcased the Word Rain visualisation technique on text comparison and dictionary development tasks, and we have also performed a user study (Skeppstedt et al., 2024). In addition, we have used the Word Rain visualisation technique in a digital history project to explore longitudinal changes in a temporal corpus (Skeppstedt & Aangenendt, 2024; Skeppstedt et al., 2025).

3 Examples of graphs generated with the Word Rain web service

In Figure 1, we have used the Word Rain web service to generate three different word rains with the top 300 words/bigrams, from three different corpora: the upper from the EuroParl-UdS corpus (Karakanta et al., 2018), spanning the years 1999-2017, the middle (House of Commons) and lower (House of Lords) from the British part of the ParlaMint corpus (Erjavec et al., 2023), only for the year 2017.

The Word Rain service removes stop words for the language of the texts, which is specified by the user before the word rains are generated. For the English texts used for the example, the stop word list employed by the web service used is the “English” list from NLTK (Bird, 2002). The English word embedding model has been retrieved from the NLPL word embeddings repository¹ (number 40, English CoNLL17 corpus, Word2Vec Continuous Skipgram).

The three graphs have been generated as one series of graphs, sharing the same word embedding projection on the x-axis. Word position on the x-axis can thereby be used for comparing the three word rains. As prominence measure term frequency is used.

We can, for instance, clearly see that *government* is prominent in both the House of Commons and House of Lords corpora. Since the word rains share the the same x-axis projection, we can also look in the European Parliament word rain at the same x coordinate, and if we look closely enough, we will find the same word there, but considerably smaller.

On the other hand, we can look at a very prominent word in the House of Lords corpus: *noble*. If we look in the European Parliament and House of Commons word rains, we will not find this word among the top 300 visualised. We will, however, find words that are used similarly nearby, such as *hon* (short for Honourable, title of member of parliament) in the House of Commons, and *mr* in the European Parliament. The Word Rain technique, thus, not only supports us in comparing graphs on an individual word level, as the for the word *government*. We can also make a comparison on a semantic category level: All three texts contain words denoting titles, but which titles are used differ between the three corpora. We can also detect clusters of words that only exist among the top 300 most prominent words in one of the corpora. For instance, the House of Lords corpus has a cluster of words denoting education (in violet, e.g., including the words *student/education/universities/academic*), which is only represented by the word *education* in the other British corpus, and not at all represented in the European Parliament corpus. The European Parliament data instead has a cluster of prominent words referring to EU and Europe (also in violet, e.g., including *European/EU/commission/council*). This word cluster also occurs in the graphs for the other two corpora, but it is not at all prominent in the other two graphs.

The discussion of differences and similarities between the graphs also exemplifies how the word positioning supports a manual, semantic categorisation of prominent words. We have, for instance, mentioned that prominent words belong to categories such as *titles*, *education* and words related to *EU/Europe*, i.e., categories that are easy to detect in the graphs thanks to the semantically motivated word positioning. We have also seen how the semantic positioning guides us to zooming in into interesting areas in the graph. For instance, the word *students*, in the House of Lords corpus and a corresponding cluster of many bars in the semantic area close to this word, made us zoom into this area, where we found more words on the topic of education. Thereby, in contrast to a traditional word cloud, there is a point of also including

¹<https://vectors.nlpl.eu/repository/>

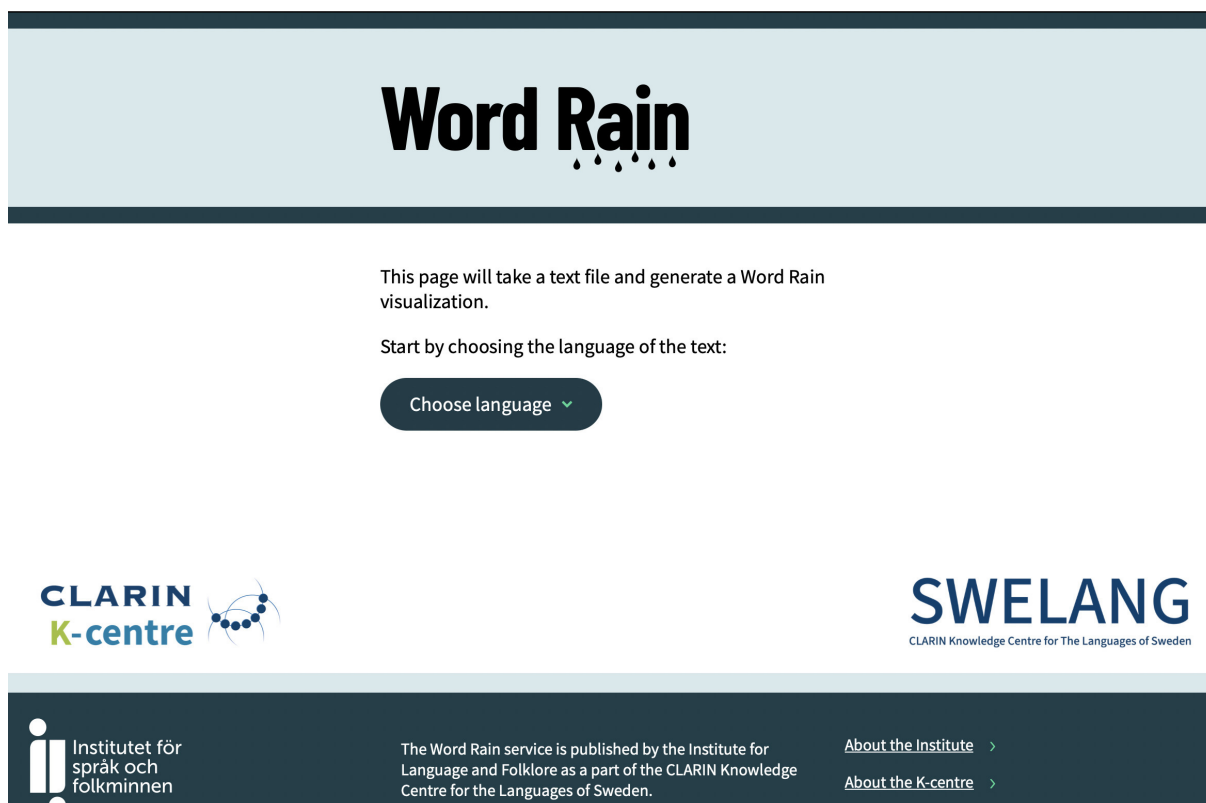


Figure 3: The Word Rain service web site at <https://wordrain.isof.se/>.

words displayed with a small font, as they can be reached through their more prominent semantic neighbours. Figure 1 also exemplifies how the coloured bars above the words give a sense of where there are prominent words, without the bias of word length.

Figure 2 shows a word rain visualisation of the same three corpora, but the 600 most prominent words are instead included in the graph. Note that this graph, thereby, shows another series of word rains, with a different semantic projection on the x-axis than what was used for the graphs in Figure 1. The same type of analysis, as exemplified for the graphs showing the top 300 words, can be conducted on the word rain series containing the top 600 words.

Together, these properties of Word Rain mean that it can be meaningful to dig deeper, and use the graph for actual exploration and analysis, as opposed to traditional word clouds, where the information content in practice is not greater than a simple sorted list of words.

It should, however, be noted that the main goal of the Word Rain visualisation technique is still the same as that of a sorted word frequency list or of a word cloud, i.e., to provide an overview of the text content by extracting and displaying its most prominent words. The goal of the Word Rain visualisation technique is, thus, *not* to represent the original word embeddings in as much detail as possible, but to use the word embeddings to order the prominent words in a meaningful way. Thereby, this novel visualisation technique is able to better support the exploration and analysis of the most prominent words in a text.

4 The Word Rain web service

Word clouds can currently be generated by a large number of tools, from numerous online websites to software libraries that can be easily integrated into custom solutions. To generate word rains, on the other hand, the user has previously been required to download the Python code, install relevant packages, find (or train) a suitable language model, and finally write a small program using the Word Rain library.

The screenshot shows the 'Word Rain' web interface. At the top, the title 'Word Rain' is displayed in a large, bold, black font. Below the title, a message states: 'This page will take a text file and generate a Word Rain visualization.' The user is prompted to 'Start by choosing the language of the text:', with a dropdown menu currently set to 'English'. Below this, the user is asked to 'Then choose number of words to plot:', with a dropdown menu set to '300'. There is a link for 'Advanced settings'. Under 'Upload either:', two options are listed: 'a text file (plain text, UTF-8)' and 'a Word .docx file'. An 'Upload file' button is located below these options. The footer contains logos for 'CLARIN K-centre' and 'SWELANG' (CLARIN Knowledge Centre for The Languages of Sweden), along with the 'Institutet för språk och kommunikation' logo and a link to 'About the Institute'.

Figure 4: After the user has chosen a language, the basic configuration options for the text processing are shown.

This means that, even though we believe the Word Rain technique is a more suitable visualisation for many applications, most people would not have the time or technical know-how to generate a word rain from their data.

We have therefore collected everything needed for generating word rains in a web-based service, where the user can just upload one or more text documents and choose parameters for the visualisation. The service is both available as a web site at <https://wordrain.isof.se/> and as open source code that can be easily deployed in standard web server environments.² At the moment, the instance deployed at wordrain.isof.se offers language models for Swedish, English, Finnish and Yiddish.

In the current version, the published code only supports plain UTF-8 text files. In order to support different file types without changing the code, the service provides a plug-in architecture where additional Python modules can be specified. Each plug-in has the opportunity to recognize and extract text from a document file format. The above-mentioned web site uses this plug-in architecture to support .docx (OOXML) files.

The web server code uses the main Word Rain Python library.³ We made a few adaptations and additions to this library when developing the Word Rain web service. One of these regards how the word embedding model files are read. In the example cases, which we provide in the Word Rain code repository, the word embedding model files are read using the Gensim library (Rehurek & Sojka, 2011). However, the functionality provided by Gensim reads the whole model into memory. Since we aim to support several languages at once in the web server, this can quickly use a lot of memory. Word Rain only reads vectors for the words that are actually plotted, meaning the performance of each access is not critical. We therefore developed our own model reading library that initially only reads an index of the words into memory, and then reads each vector from the model file as needed.

Apart from the selection of a language model, the parameters for configuring the word rains can be grouped into two categories: parameters controlling the *text processing*, and parameters controlling the

²The source code for the web service is available at <https://github.com/sprakradet/wordrain-service>

³Available at <https://github.com/CDHUppsala/word-rain>

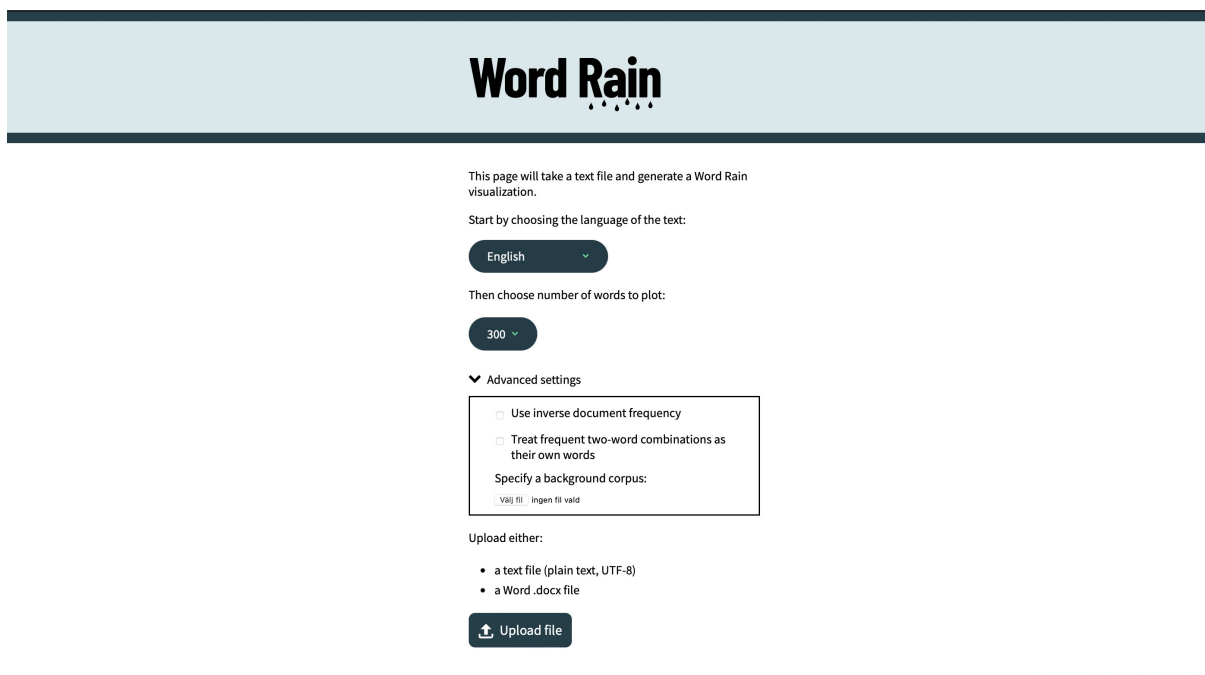


Figure 5: More advanced settings for the text processing.

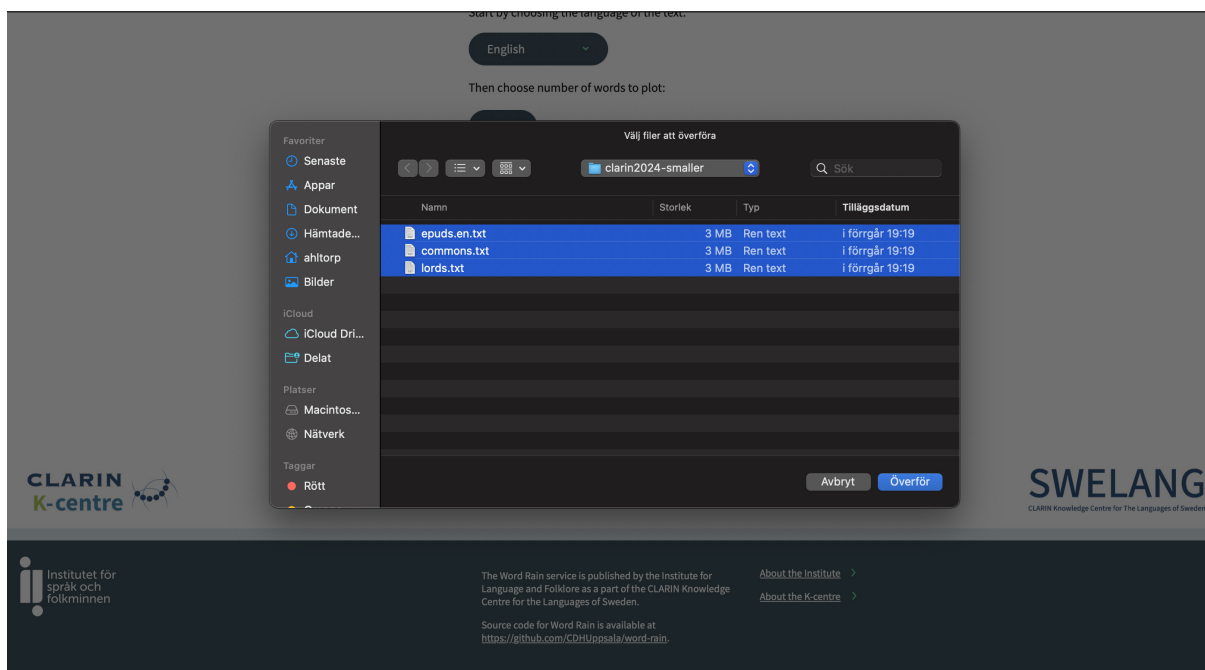


Figure 6: Choosing the texts to visualise.

their own words

Specify a background corpus:

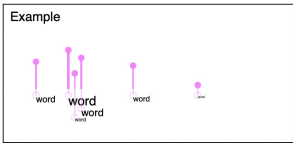
Upload either:

- a text file (plain text, UTF-8)
- a Word .docx file

Uploaded

Now choose how you want the word rain to look by configuring how quickly you want less common words to become smaller (word size fall-off) and how tall the vertical bars should be (bar height). Then click the "Draw word rain" button to see the result. Change the settings and redraw the word rain until you are satisfied.

Example




Word size fall-off: 1

Bar height: 30%

CLARIN

K-centre



SWELANG

CLARIN Knowledge Centre for The Languages of Sweden

Figure 7: Choosing configuration for the graphical presentation.

graphical presentation. Examples of text processing parameters are whether to use inverse document frequency and/or a background corpus for prominence calculations, the maximum size of n -grams that should be treated as one term, and the desired number of words to display. Graphical presentation parameters include how much space to dedicate to the vertical bars, and how sharp the drop-off in font size should be. In the following paragraphs, we will practically show how to configure the different parameters.

Figure 3 shows the information presented when the web page of the service is first loaded. When the user has chosen one of the languages offered by the deployed instance of the web service, the one basic parameter controlling the text processing becomes visible. As shown in Figure 4, this parameter consists of the number of words to include in the plot. The default value is the top 300 most prominent words, but the user can also choose to include the top 600 most prominent ones.

It is also possible to select more advanced settings for the text processing carried out. Figure 5 shows the three more advanced user settings currently available: i) To use the term frequency–inverse document frequency as the prominence measure instead of the standard term frequency measure, ii) to include prominent bigrams in the graph, and iii) to use a background corpus for the inverse document frequency calculations.

After the configuration for the text processing has been carried out, the text(s) to visualise can be uploaded. This is done by pressing the “Upload file” button, which results in a dialogue box for choosing files. This is illustrated in Figure 6, and as shown in the figure, it is possible to select multiple files. When the files have been uploaded, a progress bar is displayed to the user while the top n most prominent words are extracted from each of the uploaded texts, and a t-SNE projection is calculated from the matrix of word embeddings representing the top n words in all texts.

After the text processing has been carried out, the user can select configuration parameters for the graphical presentation. There are two parameters to set: i) “Word size fall-off”, and ii) “bar height”. The “word size fall-off” rate governs how quickly the font size of the less prominent words are to decrease in the graph. “Bar height” decides the height of the vertical bars. The value for these configuration options are set by using sliders, as shown in Figure 7.

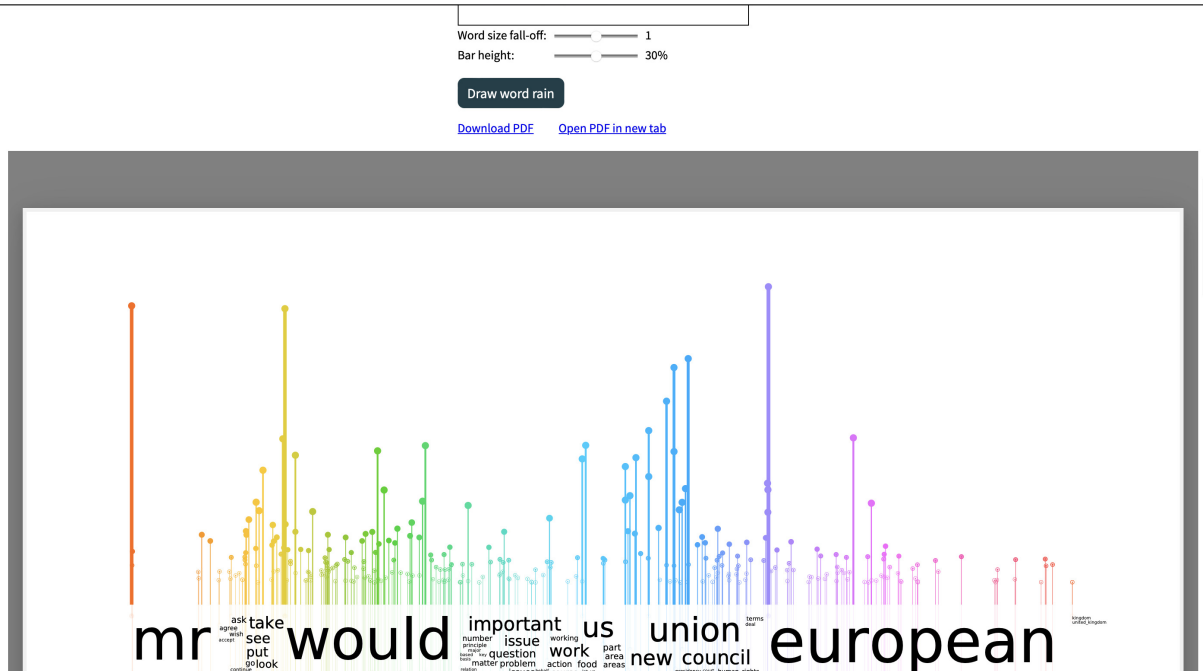


Figure 8: The word rains have been generated.

By pressing the “Draw word rain” button, the user can generate word rains with the configuration options chosen. The user is then again presented with a progress bar, while the graphs are being generated. When the graphs have been produced, they are then shown to the user, as in Figure 8. The user can then either change the configuration for the graphical presentation and generate new word rains, or – if satisfied with the design of the word rains – download the PDF-file generated.

5 Conclusions and future work

Providing an easy-to-use tool for generating word rains is crucial to widespread adoption. Many potential users do not have the technical skills required to run the Word Rain Python code library. Others do not have the time to use a code library for evaluating a new visualisation technique, even if they would eventually want to set it up themselves, should they decide to use it in their workflow. The tool is therefore a welcome addition to the SWELANG CLARIN K-centre repertoire, especially for the K-centre’s focus: the languages of Sweden. Offering it for English as well from the start makes it useful for prototyping Word Rain use from the whole CLARIN user community.

Future work includes the addition of more configuration parameters. We do not aim to include all of the configuration options offered by the Python code library into the web service, but we believe there are a number of additional configuration options that might be useful to add to the service, and which would not make the service too complex to use. We have, for instance, received requests to add the possibility of uploading a user-defined stop word list to the web service. We also plan to add support for more languages, starting with the Swedish national minority language Meänkieli.

There are also a number of technical improvements that might be relevant to implement. For instance, processing large corpora uses large amounts of memory for the calculation of word frequency. This could probably be optimised, by for example running a pre-processing pass eliminating all words under a certain frequency.

Acknowledgments

The development of the Word Rain web service is funded by the Swedish Research Council: Swe-CLARIN/The National Language Bank of Sweden (2017-00626).

The development of the Word Rain Python library code is funded by the Swedish Research Council: Swe-CLARIN/The National Language Bank of Sweden (2017-00626), Huminfra (2021-00176) and InfraVis (2021-00181).

References

- Barth, L., Kobourov, S. G., & Pupyrev, S. (2014). Experimental comparison of semantic word clouds. In J. Gudmundsson & J. Katajainen (Eds.), *Experimental algorithms* (pp. 247–258). Springer International Publishing. https://doi.org/10.1007/978-3-319-07959-2_21
- Bird, S. (2002). NLTK: The natural language toolkit. *Proceedings of the ACL Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics*.
- Burch, M., Lohmann, S., Beck, F., Rodriguez, N., Di Silvestro, L., & Weiskopf, D. (2014). RadCloud: Visualizing multiple texts with merged word clouds. *Proceedings of the International Conference on Information Visualisation*, 108–113. <https://doi.org/10.1109/IV.2014.72>
- Cao, N., & Cui, W. (2016). *Introduction to text visualization* (Vol. 1). Atlantis Press. <https://doi.org/10.2991/978-94-6239-186-4>
- Cui, W., Wu, Y., Liu, S., Wei, F., Zhou, M., & Qu, H. (2010). Context-preserving, dynamic word cloud visualization. *IEEE Computer Graphics and Applications*, 30(6), 42–53. <https://doi.org/10.1109/MCG.2010.102>
- Diakopoulos, N., Elgesem, D., Salway, A., Zhang, A., & Hofland, K. (2015). Compare Clouds: Visualizing text corpora to compare media frames. *Proceedings of the 2015 IUI Workshop on Visual Text Analytics*.
- Erjavec, T., Kopp, M., Ogrodniczuk, M., Osenova, P., Agirrezabal, M., Agnoloni, T., Aires, J., Albini, M., Alkorta, J., Antiba-Cartazo, I., Arrieta, E., Barcala, M., Bardanca, D., Barkarson, S., Bartolini, R., Battistoni, R., Bel, N., Bonet Ramos, M. d. M., Calzada Pérez, M., ... Fišer, D. (2023). Multilingual comparable corpora of parliamentary debates ParlaMint 4.0 [Slovenian language resource repository CLARIN.SI]. <http://hdl.handle.net/11356/1859>
- Herold, E., Pöckelmann, M., Berg, C., Ritter, J., & Hall, M. M. (2019). Stable word-clouds for visualising text-changes over time. *Digital Libraries for Open Knowledge*, 224–237. https://doi.org/10.1007/978-3-030-30760-8_20
- Hicke, R. M. M., Goenka, M., & Alexander, E. (2022). Word clouds in the wild. *Proceedings of the IEEE Workshop on Visualization for the Digital Humanities*, 43–48. <https://doi.org/10.1109/VIS4DH57440.2022.00015>
- Karakanta, A., Vela, M., & Teich, E. (2018). EuroParl-UdS: Preserving and extending metadata in parliamentary debates. *Proceedings of the LREC 2018*.
- Lee, B., Riche, N. H., Karlson, A. K., & Carpendale, S. (2010). SparkClouds: Visualizing trends in tag clouds. *IEEE Transactions on Visualization and Computer Graphics*, 16(6), 1182–1189. <https://doi.org/10.1109/TVCG.2010.194>
- Liu, X., Shen, H.-W., & Hu, Y. (2015). Supporting multifaceted viewing of word clouds with focus+context display. *Information Visualization*, 14(2), 168–180. <https://doi.org/10.1177/1473871614534095>
- Rehurek, R., & Sojka, P. (2011). Gensim–python framework for vector space modelling. *NLP Centre, Faculty of Informatics, Masaryk University, Brno, Czech Republic*, 3(2).
- Skeppstedt, M., & Aangenendt, G. (2024). Using the Word Rain technique to visualize longitudinal changes in periodicals from the Swedish Diabetes Association. *Proceedings of the Workshop on Visualization for Natural Language Processing*. <https://doi.org/10.2312/vis4nlp.20241132>

- Skeppstedt, M., Ahlthorp, M., Aangenendt, G., & Söderfeldt, Y. (2025). Further developing the Word Rain text visualisation technique in a digital history project. *Digital Humanities in the Nordic and Baltic Countries Publications*, 7(2). <https://doi.org/10.5617/dhnbpub.12292>
- Skeppstedt, M., Ahlthorp, M., Kucher, K., & Lindström, M. (2024). From word clouds to Word Rain: Revisiting the classic word cloud to visualize climate change texts. *Information Visualization*. <https://doi.org/10.1177/14738716241236188>
- Wang, Y., Chu, X., Bao, C., Zhu, L., Deussen, O., Chen, B., & Sedlmair, M. (2018). EdWordle: Consistency-preserving word cloud editing. *IEEE Transactions on Visualization and Computer Graphics*, 24(1), 647–656. <https://doi.org/10.1109/TVCG.2017.2745859>
- Wu, Y., Provan, T., Wei, F., Liu, S., & Ma, K.-L. (2011). Semantic-preserving word clouds by seam carving. *Computer Graphics Forum*, 30(3), 741–750. <https://doi.org/10.1111/j.1467-8659.2011.01923.x>
- Xie, L., Shu, X., Su, J. C., Wang, Y., Chen, S., & Qu, H. (2024). Creating emordle: Animating word cloud for emotion expression. *IEEE Transactions on Visualization and Computer Graphics*, 30(8), 5198–5211. <https://doi.org/10.1109/TVCG.2023.3286392>
- Xu, J., Tao, Y., & Lin, H. (2016). Semantic word cloud generation based on word embeddings. *Proceedings of the IEEE Pacific Visualization Symposium*, 239–243. <https://doi.org/10.1109/PACIFICVIS.2016.7465278>