TRACING AND VISUALIZING MULTILINGUAL DIACHRONIC SEMANTIC CHANGE WITH CONTEXTUALIZED EMBEDDINGS

Mathilde AGUIAR
mathilde.aguiar1@etu.univ-lorraine.fr
Averie Ho Zoen SO
ho-zoen.so4@etu.univ-lorraine.fr
Duy Van NGO
van-duy.ngo9@etu.univ-lorraine.fr
Scott TANKARD
scott-propst.tankard7@etu.univ-lorraine.fr

ABSTRACT

Lexical semantic change is a topic that attracts interest from a wide range of audiences, ranging from historical linguists to the general public. This project aims to answer a question like "how does the word *gay* change from meaning *merry* to *homosexual*?" Traditionally, the answers may be answered by etymological accounts or by citing historical events. By temporally adapting pre-trained, contextualised word embeddings with the mBERT model and clustering, we present a tool that can visualise this kind of semantic change over time. We find that our multilingual model achieves comparable performance in semantic change detection compared to previous approaches, and additionally, that multilingually fine-tuned mBERT is beneficial to such a task. We present several case studies via our visualisation component and discuss wider implications for future research.

1 Introduction

The study of semantic change is a field of research with a long history in linguistics. Understanding how languages change can have significant implications for various fields of linguistic research, including psycholinguistics, sociolinguistics, semantic theories and language reconstruction. Moreover, semantic change is often a topic that gathers interest from outside of linguistics, as the explanations of these changes often involve historical, social and political factors, and can have practical implications for fields such as legislation [1] and political science [2].

Traditionally, studies of semantic change rely on a few handpicked examples by linguists in an attempt to extract regularities and patterns that can be generalised to different lexical items. Advances in computational approaches to solving language tasks such as translation, summarization and dialogue have concurrently led to the development of word embeddings – a numerical representation of word meanings in relation to all other words in a language [3]. This means that a large-scale, quantitative method is possible to validate and extend the various hypotheses from historical linguistics.

In recent years, models of contextualised word embeddings such as BERT [4] have achieved major breakthroughs, achieving state-of-the-art performances in many language tasks. Compared to previous approaches such as word2vec [3], contextualised word embeddings have several advantages, which includes the ability to capture multiple senses of the same lexical item as well as pre-trained multilingual models. The SemEval task of Unsupervised Lexical Semantic Change Detection (LSCD) in 2020 [5] has effectively boosted the popularity in solving this task, especially by providing standardised gold datasets across 4 languages – English, German, Swedish and Latin. The outcome of the task also serves as one of the most comprehensive systematic comparison across different approaches to detecting lexical semantic change. In particular, the finding that various approaches via static embeddings (one embedding per orthographic form) performing better than those via contextualised embeddings (one embedding per occurrence) was unexpected, as contextualised embeddings have been shown to outperform static embeddings in many other NLP tasks, making LSCD a rare exception to the rule at the time [6].

Despite this, conceptually, there are various reasons why we might prefer contextualised embeddings for the task of modelling diachronic semantic change.

- BERT is a pre-trained language model that has been trained on a sufficiently large corpus, which eliminates the difficulties in collecting datasets to acquire semantically representative embeddings.
- BERT embeddings are adjustable with simple and inexpensive fine-tuning, which potentially allows us to study the meaning changes without an excessively large dataset.
- BERT embeddings support multiple word senses, which is helpful in learning meaning gains/losses and representing meaning more accurately.

Approaches thereafter have proposed novel methods with contextualised embeddings which have shown comparable results to static-embedding approaches on parts of the SemEval test set [6, 7]. Additionally, since we recognise that the nature and significance of lexical semantic change should not be limited to numerical presentations, we go beyond semantic change *detection* by implementing a visualisation component which allows users to explore and interpret such changes. Therefore, the current project proposes an alternative approach to utilise contextualised embeddings for the task of modelling lexical semantic change, in three main aspects:

1. the use of a pre-trained multilingual model (mBERT) to model lexical semantic change
2. an approach to fine-tune mBERT on time-specific corpora
3. a visualisation component on the web that represents such changes

This report presents the details and examples to our tool that can visualise and trace how the meanings of words change over time as represented numerically by multilingual, contextualised word embeddings. The structure of the report is as follows: in section 2, we describe our motivations and modifications to the original method by Montariol et al [7]. In section 3, we discuss other recent approaches to studying semantic change with computational methods. Section 4 describes in detail the methods of the software and section 5 presents quantitative and qualitative evaluations of the performance of our models. Section 6 gives a more detailed description of the user interface. Finally, section 7 discusses our findings and future directions.

2 Our approach

Utilising the state-of-the-art embeddings from emerging neural models such as Transformer and Recurrent Neural Network, the study of semantic representations of text has been seemingly accelerated. For instance, contextualised, pre-trained embeddings such as BERT are designed to provide context-sensitive and intuitive vector representations of words that stimulate real-life semantic meanings. With these pre-trained models trained on sufficiently large corpora, the study of semantic changes between time periods is presumed highly possible with a little amendment at the fine-tuning step. In this section, we discuss the approach in detail.

Our approach is heavily based on Montariol and colleagues [7] with some significant modifications. Montariol et al approached the task of LSCD by fine-tuning a model on a corpus of documents from across all time slices all at once, essentially domain adapting to the corpora (eg. COHA) rather than to the time-slice. The contextualised embeddings of time-specific corpora are then retrieved from the same model. At each time slice, clusters are formed by comparing the many vectors retrieved from the same word, but in different sentences. These clusters then allow for a measurement of semantic change carried out by Wasserstein Distance (WD) [8] or Jensen-Shannon Divergence (JSD) [9]. These measures are designed to measure the difference in the probability distribution, and in the case of JSD, it has been used in the task of modelling semantic change by various previous approaches [6, 10, 11]. We explain these measurements in more detail in Section 4.2.4

In the current project, we have adapted the approach such that the embeddings were fine-tuned to individual time slices instead of a time-agnostic domain. This method would be more conceptually sound to our purpose of tracing diachronic change since our goal is to reflect the changes of various related query words in the various time-specific vector spaces, beyond measuring the amount of change for each target word individually, as Montariol et al had intended. As such, by having each model per time slice our method can be potentially more flexibly adapted to include more than two time slices in the pipeline and can represent how changes happen across multiple query words. We train multiple models using a common baseline model (off-the-shelf mBERT) with time-specific datasets. For each time slice, we produce a separate mBERT model through the fine-tuning of the baseline model using the data from the aforementioned time slice. This methodology is intended to help us avoid the unnecessary adjustment of embeddings caused by training the same model using time-noisy datasets.

3 Related Work

3.1 Semantic Change Modelling

Semantic change is a complex phenomenon that has a multitude of aspects beyond change *detection*. Previous research has targeted different aspects via computational approaches. Static embeddings, in particular Word2Vec, have been popular in this line of research. For example, Rosin et al [12] focused on word relatedness (*president - Obama*), rather than similarity (*president - prime minister*) for the purpose of a time-aware search system. Szymanski [13] compared words that occupy the same location across time-specific vector spaces by extracting temporal analogies such as *"Ronald Reagan" in 1987 is like "Bill Clinton" in 1997*.

In terms of contextualised word embeddings, the widely applicable procedure of fine-tuning BERT for many NLP tasks has also introduced another wave of research. Qiu and Xu [14] investigated the hypothesis that BERT would be biased towards recent language usage since it is trained on prominently contemporary data from the internet. By fine-tuning BERT on Corpus of Historical American English (COHA), their model is designed to be more *historically balanced* and therefore gained improvements in correlating with a human-judged temporal semantic similarity task. Temporally adapting BERT has also been shown to be applicable to time frames of finer granularity, Hombaiah et al [15] incrementally trained BERT for each individual year and found that the models fine-tuned to the particular year is beneficial for tasks such as offensive tweet classification and Country Hashtag prediction (predict the associated country given a hashtag).

3.2 Lexical Semantic Change Detection (LSCD)

In the specific case of lexical semantic change, we surveyed a number of papers in the domain, and present here a selection of works which we found most relevant, some of which we considered adapting for our purposes.

The HistWords[1] project [16] is a collection of software tools and datasets, aimed at building and analyzing word vectors with a focus on change over historical time. They present quantified semantic changes in word embeddings, via experimenting on 6 datasets in 4 languages: EN, ZH, DE, FR. Time-specific embeddings are aligned with procrustes regression, and pairwise similarness is calculated using cosine-similarity. The visualisations they present, as well as the statistical laws of semantic change, were a major inspiration for and influence on our project.

In their submission to SemEval 2020 Task 1, Arefyev and Zhikov tackle the issue of word sense induction, approaching it using lexical substitution, which they apply to perform Lexical Semantic Change Detection (LSCD). Their paper employs a three-step BOS + AggloSil + DC approach involving bag of substitutes (BOS) vectors, agglomerative clustering via silhouette scoring ("AggloSil"), and selecting a "decision cluster" (DC).[2] These researchers present two main tools: a tool `BOS_AggloSil` for performing word sense induction (WSI) via clustering, and their corresponding visualisation tool `SCD_WSI_tool`.[3] We took significant inspiration from their visualisation tool presenting clusters of word meanings as well as change over time.

While contextualised word embeddings have been shown to outperform static word embeddings in general, there are unique challenges when using them for detecting lexical semantic change. The first attempt to use contextualised embeddings for LSCD was Giulianelli et al[10]. They had dealt with one of the main problems, but also a benefit, of using contextualised embeddings, which is the challenge of tracking senses over time. Since embeddings may exist in different spaces in different models, the different senses have to be tracked to deal with multi-sense usage in a way that can be followed across time slices. As such, Giulieanelli et al used clustering to first tag the individual senses and then compare how the vectors changed across the original BERT and a time-specific BERT model.

[1] `https://github.com/williamleif/histwords`

[2] For more detail, see this quotation from their paper: "For a particular target word its occurrences in old and new corpora are collected, and lexical substitutes are generated for each of them. WSI [word sense induction] is performed by clustering bag-of substitutes (BOS) vectors with agglomerative clustering employing silhouette score to select the number of clusters (AggloSil). Finally, we search for a decision cluster (DC), i.e. a cluster that has large number of occurrences from one corpus and small from another, and predict semantic change if such cluster exists." [17]

[3] Their main repo: `https://github.com/DeadBread/SCD_WSI_tool`; corresponding visualization tool (web app): `https://github.com/DeadBread/BOS_AggloSil`. Note that in the process of examining their codebase, we created a requirements.txt and conda environment with each problem ironed out (specific versioning issues resolved by testing different python versions), as well as call graph diagrams produced with code2flow, which may be of use to others who wish to examine their code or reproduce their results.

4 Methods

4.1 Data

We used the train and test data from the SemEval-2020 Task 1 – Unsupervised Lexical Semantic Change Detection [5]. Out of the four languages in the task (English, German, Latin, Swedish), we fine-tuned our models on English and German data, this is because these two languages have similar time frames in both time periods, in which C_1 is from the 1800s and C_2 is mostly from the 1900s. To ensure that our models are fine-tuned to represent meaning change realistically, we use the "token" version of the SemEval datasets, instead of the "lemma" version that were provided. Additionally, the sentences in the datasets were already cleaned and shuffled. Table 1 shows the data statistics in detail.

	C_1			C_2		
Lang	**corpus**	**period**	**line count**	**corpus**	**period**	**line count**
English	CCOHA	1810-1860	234,917	CCOHA	1960-2010	331,055
German	DTA	1800-1899	260,200	BZ+ND	1946-1990	350,480

Table 1: Summary of the English and German training data. German data has been downsampled to 10% of the original data in order to balance with English data.

4.1.1 Training Data

English obtained from the Clean Corpus of Historical American English (CCOHA) [18, 19]. This corpus is created with the aim to include a wide variation of genres, which include TV/ movies, fiction, popular magazines, newspapers and non-fiction books [4].

German a combination of news corpora. Data in C_1 is from the DTA corpus [20] and C_2 is from BZ and ND corpora [21, 22]. When fine-tuning mBERT, we downsample the German corpora to 10% of the original such that the sizes are comparable to the English corpora.

4.1.2 Test Data

The SemEval test set is a list of target word lists and the corresponding amount of semantic change. For English, there were 37 target words. Pairs of word usage (ie. one target word in the context of two sentences) from C_1 and C_2 were judged by annotators. The judgements were made on a four-point scale: (1) Unrelated, (2) Distantly related, (3) Closely Related and (4) Identical. The details of how the overall scores were calculated can be found in Schlechtweg et al [5].

4.2 Pipeline

This section focuses on the backend pipeline component, involving corpus preparation, fine-tuning the models, extracting embeddings of interest from the fine-tuned models, running measurement metrics on the extracted embeddings comparing time slices, and in the final step, a dual output of evaluation metrics (for automatic evaluation against gold-standard data), and data files to be fed into the visualisation component. The visualisation component itself is discussed in another section.

We took as starting point for our pipeline the scripts in the scalable_semantic_shift repository. This repository provides a number of tools relevant to our project, as it was a similar project in scope and goal, but with some key differences. In particular, they fine-tune a single mono-model rather than separate models per time slice; additionally, theirs is a research focus with hardcoded defaults, rather than a toolkit/application focus with UI. We take the codebase that they made available, and build on their work in a different direction. These are the relevant set of scripts from this codebase that we focused on wrapping and adapting to fit our goals:

- `build_coha_corpus.py`
- `fine-tune_BERT.py`
- `get_embeddings_scalable.py`
- `measure_semantic_shift.py`
- `evaluate.py`

[4] https://www.english-corpora.org/coha/

The main output of our work, in terms of software deliverables for the pipeline component, is threefold:

- A set of wrappers, helper functions and command-line tools, tying together the various components of the pipeline, in `g5_tools.py`. We also implement extensive and detailed logging (useful for reproducibility of experiments, unattended runs, troubleshooting, and measuring time and computing resources spent), capture of subcommand output, checksumming of inputs and outputs, and other mechanisms, in a flexible and modular framework.
- Code for both filtering a corpus based on query keywords of interest, as well as downsampling based on desired corpus size limits, in `reduce_corpus.py`.
- Code for running measurement metrics on the extracted embeddings, merging the pickle files for different time slices, in `measure_semantic_shift_merged.py`, and for producing semantic distance measurements useful for the visualisation component, in `measure_semantic_shift_visualisation.py`

4.2.1 Corpus Preparation

When training on two languages, we concatenate the English and German corpora. We employ random downsampling on the German data in order to bring it in line with the quantity of English data (the raw data, as provided by SemEval, is roughly 10 times as large for German as for English). As the SemEval dataset is already significantly preprocessed, the amount of preprocessing required is very moderate. We use the tokenized datasets from within the SemEval archive (`token/`), not the lemmatized ones (`lemma/`).

In addition, during prototyping and testing, we also implemented slicing of the COHA sample corpus[5] into per-year time slices. We use this codepath for testing, and in future we plan to adapt it to other corpuses, and larger corpuses, for use in production. We maintain our preprocessing step as a fully modular component within the pipeline, with the aim of integrating other languages and corpuses of various types, with more fine-grained time-period granularity, in future work.

4.2.2 Fine-tuning BERT Model

BERT is pre-trained on contemporary corpora (e.g. Wikipedia) which do not appropriately reflect language use in historical times. Conceptually, our attempt to alleviate this problem is to fine-tune (further continue pre-training from the standard published checkpoint) mBERT on time-specific corpora, in an unsupervised manner, on the masked language modeling task (MLM). We segment corpora from each language for each time period, then we train mBERT embeddings for each of these time periods multi-lingually.

We fine-tuned a model from bert-base-multilingual-cased model (mBERT) provided by Huggingface [6] at each time frame and with two variations. In the first variation, the models are only trained on English data; in the second variation, the models are trained on a concatenated file of both German and English data. This results in 4 models, where each model is trained for 5 epochs on a GPU. We used both a batch size of 7 and a batch size of 4, when training on different machines, due to GPU RAM size limitations. In each variation, the word embeddings are then extracted from the respective C_1 and C_2 models, resulting in two sets of results.

In our project, we have thus trained on two languages simultaneously: English and German. In addition, our framework is general and flexible enough to be adapted to further languages in future. In particular, the fact that we plan to apply narrower time-slicing (for example granularity of one year) when applying our methods to larger data in future work, will allow us to effectively tackle large datasets in pieces, using an array of models, training on a reasonably sized quantity of data at a time. We believe that this approach is scalable and generalizable.

4.2.3 Embeddings Extraction

We extract, from the fine-tuned models we produce, contextualised embeddings for query words. The results of these embeddings are saved into a pickle file format, following the work of [7]. The extraction step is premised on a query wordlist, and the output contains a list of embeddings (each in the form of a `numpy` array) for each query word, as well as references to where they appear in the source corpus. As tokenizer we use the BertTokenizer from HuggingFace, corresponding to the pre-trained model we are using (bert-base-multilingual-cased). We disable conversion to lowercase, for all languages.

In order to reduce the time spent on extracting word embeddings, we employ an additional step to filter only the sentences containing at least one word stems that match the list of target word stems. Specifically, we retrieve the word

[5] `https://www.corpusdata.org/coha/samples/text.zip`

[6] https://huggingface.co/bert-base-multilingual-cased

stems from the language-specific stemmers provided by NLTK [7]. To ensure that sentences are not filtered out due to morphological variants, we further cut the stems that are longer than 4 characters such that all stems are at most 4 characters long, we find that this pattern generally produces a relatively balanced number of remaining sentences for both English and German.

This step of filtering per query words is not used for the training of the models; the models are trained on a full dataset, and thus have access to the full range of linguistic content to generalize from. This step of filtering is used only on the querying and extracting of word-specific embeddings from the finished model. In this way, we gain the benefit of a large dataset for training, and avoid the downsides of dealing with large data when it is not strictly necessary (this implies gains in time as well as less use of computing power).

4.2.4 Measuring Semantic Shift

We exploit the extracted word embeddings in two ways. First, to apply clustering and calculate change in distribution via Wasserstein Distance (WD) and second, to compute relations with pre-defined neighbours in relation to a target word. We describe them as follows:

1 **clustering and measuring shift in a probability distribution**: Using the time-based `pickle` files generated from the BERT models, we obtain the embeddings of the target words and the occurrences of the words with slice markers. After clustering the vector representations of a word to obtain the senses, we count the occurrences of each sense and compute the distribution of each sense for the computation of metrics: JSD and WD. For KMEANS, we train KMeans models of `k` over the embeddings and eventually combine all the slice-based clusters into one for computing the differences between the slices. The results of this measurement is exported to a `.csv` file.

2 **comparing distance from neighbours across time**: We use the same word embeddings to compute the changes of neighbouring words of the same word sense across time slices. We first retrieve the most frequent sense of the word with its embeddings and compute the distances from that word to the rest of the vocabulary. The distance is subsequently sorted in order to obtain the `n` closest neighbours of the word using either of the JSD or WD metrics. The outcome of this exploratory approach is then visualised on our web interface.

The generated `.csv` files are stored in our server for a more efficient data retrieval using the web application.

4.2.5 Evaluation

After getting a score of semantic change from our two models, we evaluate the outputs against the human-annotated test set of semantic change provided by SemEval 2020 Task 1 by Spearman's rank correlation, which only takes into account the rank-order across the list of target words and not the absolute values of change. We only evaluate our models in the context of subtask 2 for graded predictions since we are primarily interested in the relative, fine-grained semantic differences and not in binary change detection.

5 Results

5.1 Automatic Evaluation

Table 2 shows the Spearman's rank correlation of our system's performance against the human-annotated test set, in comparison to a few other notable systems: Schlechtweg et al [23], the previous state-of-the-art semantic change detection performance via static embeddings; Pomsl and Lyapin [24], who was the winner of the SemEval task which is also based on static embeddings, and Montariol et al [7], the approach on which our pipeline is mainly based. Our best model does not outperform the original implementation by Montariol et al [7], although the performance is comparable with Pomsl and Lyapin and outperformed Schlechtweg et al. We observe that our scores slightly improve when the model is trained on multilingual data (EN + DE) compared to the monolingually trained model (EN only).

Table 3 shows the system performance using the various types of clustering methods experimented on in [7], measured by Wasserstein Distance (WD). Among our models, the best performance for English is a k-means model with 5 clusters. This does not replicate the best English model from Montariol et al [7], which is one with affinity propagation (AP). Compared across our two models, we observe that k-means 5 consistently and significantly outperforms the two other methods - k-means 7 and Affinity Propagation (AP), in this order, while the original implementation did not report a single best clustering method across different models.

[7] https://www.nltk.org/api/nltk.stem.html

Model	Embeddings type	English
Schlechtweg et al (2019) [23] [8]	SGNS	0.321
Pomsl and Lyapin (2020) [24] [9]	SGNS	0.422
Montariol et al (2021) [7]	BERT	**0.456**
Our model (EN), k-means 5	mBERT	0.408
Our model (EN + DE), k-means 5	mBERT	0.421

Table 2: Spearman's rank correlation with human-annotated semantic change, based on 37 target words for English.

Model	k-means 5	k-means 7	AP
Montariol et al (2021)	0.360	-	**0.456**
Our model (EN)	0.408	0.384	0.354
Our model (EN + DE)	0.421	0.383	0.340

Table 3: Spearman correlation with WD as measurement, across three different methods of clustering. AP = Affinity Propagation

5.2 Case studies

To have an overview of the results of our model paired with our Web interface, we selected a few words as case studies.

Gay

Figure 5 shows the results obtained for the word `gay` and rendered in the table of our user interface.

The following results are obtained by calculating the cosine similarity of each of the words in the column `Word` with the target word `gay`, for each studied time period. We observe that *cheery* was the closest word of *gay* during the time slice 1810-1860, meanwhile *merry* was the furthest. However during 1960-2010, it became close to *homosexual*, word that was not appearing in the top 10 closest word in the previous time slice. We also notice that the word *cheery* disappeared from the top 10 in the second time slice.

We can visually check these results via our scatter plot in Figure 6. As stated before, the closest point to the circle labelled `gay` is *cheery* for the first time slice and *homosexual* for the second one. We can also observe that the *homosexual* dot is way much closer than the other dots for the same time period. These results are similar to the ones obtained in [16], where the authors found that gay was semantically closer to *daft* or *flaunting* in the early 1900s and progressively derived to the *homosexual*, *lesbian* in the 1990s.

Pink

The second case study is the word `pink` from the English corpus. As the previous one, we display the different distances in a table 7 and in a scatter plot 8. Here, we observe that the closest word in the first time slice is *girl* and the furthest was *father*. The word seems strongly related to the female gender. However in the second time slice, *girl* became the furthest word meanwhile *gay* became the closest one, followed by *jolly* and *merry*.

German

For this case study we will compare the results obtained for `german` (from COHA) and `deutsch` (from DTA).

For the English `german`, as displayed in Figure 9, the closest word was *military*. However it was relatively far with a distance of 0.36 in the first time slice, meanwhile in the second slice it became closer with a score of 0.206. We can also notice that *war* and *dictatorship* became closer, especially for *dictatorship* that was the furthest word of the first time slice. We can clearly see the impact of the two World Wars on the perception of Germany by American people.

For the German `deutsch`, as in Figure 10, unlike in the COHA dataset, the top 10 similar words are not related to war or political regimes, but to other neighboring countries. *Diktatur* still appears in the second time slice but is surprisingly further than *Demokratie*.

It is also worth noticing that the English dataset, COHA, is from an American English point of view. Same goes for the DTA dataset that is from a German point of view. That might explain a few of the surprising results.

6 User Interface

For enhancing the accessibility of our project, as well as sharing our achievements with the public, we have developed a web application, reachable at `https://opensemshift.herokuapp.com/` to deliver the services. In this section, the architecture, design, implementation, and maintenance dimensions of the web application will be brought into discussion.

6.1 General purpose

In short, the web application allows the user to visualise the semantic changes of words with just a query away. Given a user-defined lexical unit, our web application demonstrates the semantic changes of the lexical unit across the desired or available (if not specified) time period by querying the database and plot the changes, scientifically. Furthermore, the website also displays the important information related to the lexical unit such as the datasets used to generate the embeddings of the word(in the `Data` tab), the methods used (in the `Methodology` tab), elevating the transparency of the system through a friendly and customisable experience.

On the other hand, we provide the public with free access to the source code of our web application [10]. On top of that, our website is committed to not gathering users' personal information without consent, including but not limited to the non-essential cookies according to Heroku's terms and conditions.

6.2 Preliminary study

At the current stage of development, our alpha release of the website is for demonstration purposes only. The current web application provides the visitors with the following features:

- Lexical unit/word querying
- Visualising semantic changes using plots and/or tables
- The embedded historical events in the visualisation
- The listings of available words in our database
- The information about our research methodologies
- The information about the datasets we used

6.3 Open SemShift architecture

The web application is designed as a complete application with two separate components: front-end and back-end. In the subsections below, we discuss the architectural aspects of this web application, beside conveying the technical justifications for the design and implementation.

6.3.1 Overall workflow

Our web application uses the outputs from the standalone module *measure_semantic_shift.py* within the *Scalable Semantic Shift* block:

The results from the `Scalable Semantic Shift` standalone module are exported and used to render our tabular, textual, and graphical contents. Since our vocabulary size is relatively small at the current stage, we have generated the results for all the available words with corpus slices and store them in `.csv` files. These `.csv` will be used as the prototypical data source for the application's back-end. The information stored in these `.csv` files are used on-demand to deliver the measurements and related computation results to the users through various approaches, such as Bokeh plots or tables, for instance.

6.3.2 Trade-offs

During the development of our web application, we faced several challenges concerning the development of our experimental solution. Our approach using one model for one time slice induces the training and generation of several models. This method is known to be expensive regarding the hardware resources, especially storage, as the size of the model increase substantially with the increasing size of the vocabulary. With the limited resource allowance permitted by Grid5k, the scaling of this project remains a great potential. To overcome this objective difficulty we decided to

[10] `https://github.com/2022-2023-M2-NLP-Group-5/production-app`

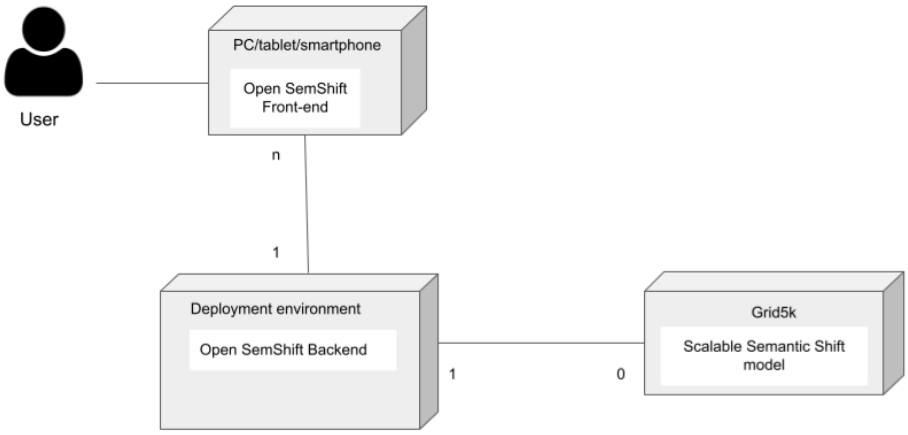

Figure 1: Overall system workflow

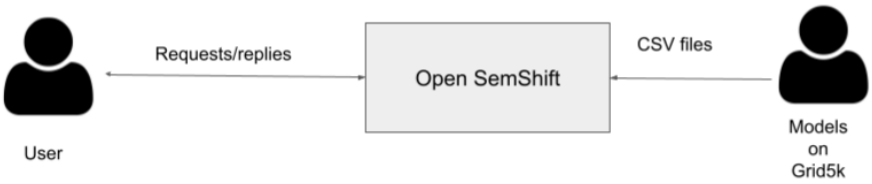

Figure 2: Overall application workflow

narrow the user queries available to a pre-defined list of words available on display in our website in the `Data` section. This implies that we do not do real-time queries on the models or any component of *Scalable Sememtic Shift*. This temporary solution helps us deliver optimistic results without performing resource-intensive tasks.

6.4 Tech stack

For the website development we used the lightweight and robust Flask framework on Python environment, paired with a couple of common libraries. The outline of the tech stack is as below:

- **Back-end**: Python Flask
- **Front-end**: Bootstrap Flask, Bootswatch, Flask WTForms
- **Visualization**: Python Bokeh

6.5 Potential future implementations

As previously mentioned, the alpha release of this website is for demonstration purposes only. Thence, the number of developed functionalities are still limited. The first improvement for further development would be the visualization. In this version, we developed 2 different means of visualization: a scatter plot with years and semantic similarities, and a tabular representation that displays the results obtained. The third mean of visualisation based on trees would be interesting with more emphasis on senses derivation.

Another aspect we find challenging is the deployment architecture. We currently use Heroku to host only our website. We have to carry these resource-intensive tasks out on Grid5k, due to the limited computation performance of a

free Heroku server. It could be ingenious to allocate different modules in dedicated and seamless production and deployment environments. However, as mentioned before, the models' components and outputs are hardware demanding, subsequently involving the additional costs we cannot afford.

Last but not less, we opted for the amendment of the graphical contents and included a few interesting historical events in the cluster visualization. From our perspective, adding more events with optimal matching of events and lexical units that are related to each other can also incubate exciting features and enhance the overall study experience.

7 Discussion

7.1 Analysis

We have presented a multilingual approach to modelling lexical semantic change along with a tool that can represent semantic change visually. By fine-tuning pre-trained, contextualised word embeddings on time-specific data, we find that mBERT can achieve similar results to other monolingual BERT approaches in the task of ranking graded semantic change in English. Notably, our multilingually trained has even improved the performance. Further, we also find that our models have consistently achieved the best results with k-means clustering and a pre-defined number of 5 clusters. Applying the fine-tuned word embeddings to our visualisation tool has shown potential for exploratory discoveries for querying lexical semantic change.

While our models achieve competitive performance in the English SemEval test set, we have not evaluated our models in German or other languages, which would be important to validate the benefits of multilingual semantic change analysis. There are several challenges that we have met in terms of the visualisation software, including the inability to query new words on the fly, and having to rely on previously extracted embeddings. While the original approach by Montariol et al [7] was designed to be scalable, it has shown not to be the case during our implementation, which requires deeper investigation in the backend. Additionally, while our approach is designed to incorporate multilingual usage, many language-specific parameters are yet to be implemented, such as filtering with appropriate lengths and whether to keep capitalised words.

Nonetheless, the finding that mBERT can perform just as well as static embeddings for semantic change analysis is exciting, as it contributes to falsifying the finding that static embeddings generally perform better than contextualised embeddings in LSCD. Moreover, our approach is novel in terms of the use of a multilingual pre-trained model for this task. Our results show potential to explore how multilingual historical data can aid this task, and more importantly, whether the diachronic semantic analysis in low-resource languages can be achieved in this way.

7.2 Future work

In our current project, we have set up with two languages, for two time periods. For future work, it would be important to extend this work to more than two time periods, since our visualisation tool is advantageous in presenting complex patterns that are not easily interpreted by numbers. Another direction is to add cleaning or standardisation procedures for orthographic differences since a change in spelling in the same words can be a common phenomenon, especially for texts from a long time ago, and in certain languages such as English.

In later work, we would also like to add additional languages into the corpuses and re-train the models. Other steps and components that we are interested to add in future include: Running experiments comparing multiple-senses *vs* single-averaged sense; testing on different types of semantic change; analyzing multiple languages in comparison to each other (e.g. evolution of Sir/Monsieur in English/French); deepening and enriching the historical event contextualization data that we currently present in the UI; and future semantic change prediction.

7.3 Conclusion

We have presented a novel approach to modelling semantic change with pre-trained, contextualised, multilingual word embeddings mBERT. To our knowledge, this is the first attempt in modelling semantic change with a multilingual model. We found that our model is competitive compared to other monolingual studies. Additionally, we found that multilingual training can be beneficial to the task of semantic change detection. Secondly, we introduce a visualisation tool that can represent how embeddings of target words change over time along with several other neighbouring words, demonstrated with several case studies of historically significant changes. We believe that our results show exciting potential for multilingual semantic change modelling. We hope that in the future this work can be extended by implementing several other important features in order for it to become a useful tool for exploratory diachronic semantic analysis.

References

[1] Elizabeth Closs Traugott. Semantic change: Bleaching, strengthening, narrowing, extension. In Keith Brown, editor, *Encyclopedia of Language and Linguistics*, pages 124–31. Elsevier, 2006.

[2] Andrey Kutuzov, Lilja Øvrelid, Terrence Szymanski, and Erik Velldal. Diachronic word embeddings and semantic shifts: a survey. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1384–1397, 2018.

[3] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26, 2013.

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[5] Dominik Schlechtweg, Barbara McGillivray, Simon Hengchen, Haim Dubossarsky, and Nina Tahmasebi. Semeval-2020 task 1: Unsupervised lexical semantic change detection. *arXiv preprint arXiv:2007.11464*, 2020.

[6] Severin Laicher, Sinan Kurtyigit, Dominik Schlechtweg, Jonas Kuhn, and Sabine Schulte im Walde. Explaining and improving bert performance on lexical semantic change detection. *arXiv preprint arXiv:2103.07259*, 2021.

[7] Syrielle Montariol, Matej Martinc, and Lidia Pivovarova. Scalable and interpretable semantic change detection. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4642–4652, Online, June 2021. Association for Computational Linguistics.

[8] Justin Solomon. Optimal transport on discrete domains. *AMS Short Course on Discrete Differential Geometry*, 2018.

[9] Jianhua Lin. Divergence measures based on the shannon entropy. *IEEE Transactions on Information theory*, 37(1):145–151, 1991.

[10] Mario Giulianelli. Lexical semantic change analysis with contextualised word representations. *Unpublished master's thesis, University of Amsterdam, Amsterdam*, 2019.

[11] Andrey Kutuzov and Mario Giulianelli. Uio-uva at semeval-2020 task 1: Contextualised embeddings for lexical semantic change detection. *arXiv preprint arXiv:2005.00050*, 2020.

[12] Guy D. Rosin, Eytan Adar, and Kira Radinsky. Learning word relatedness over time, 2017.

[13] Terrence Szymanski. Temporal word analogies: Identifying lexical replacement with diachronic word embeddings. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, Vancouver, Canada, August 2017. Association for Computational Linguistics.

[14] Wenjun Qiu and Yang Xu. Histbert: A pre-trained language model for diachronic lexical semantic analysis. *arXiv preprint arXiv:2202.03612*, 2022.

[15] Spurthi Amba Hombaiah, Tao Chen, Mingyang Zhang, Michael Bendersky, and Marc Najork. Dynamic language models for continuously evolving content. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 2514–2524, 2021.

[16] William L Hamilton, Jure Leskovec, and Dan Jurafsky. Diachronic word embeddings reveal statistical laws of semantic change. *arXiv preprint arXiv:1605.09096*, 2016.

[17] Nikolay Arefyev and Vasily Zhikov. BOS at SemEval-2020 task 1: Word sense induction via lexical substitution for lexical semantic change detection. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 171–179, Barcelona (online), December 2020. International Committee for Computational Linguistics.

[18] Mark Davies. Expanding horizons in historical linguistics with the 400-million word corpus of historical american english. *Corpora*, 7(2):121–157, 2012.

[19] Reem Alatrash, Dominik Schlechtweg, Jonas Kuhn, and Sabine Schulte Im Walde. Ccoha: Clean corpus of historical american english. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 6958–6966, 2020.

[20] Deutsches Textarchiv. Grundlage für ein referenzkorpus der neuhochdeutschen sprache. herausgegeben von der berlin-brandenburgischen akademie der wissenschaften. Berlin-Brandenburg Academy of Sciences, 2017.

[21] Berliner Zeitung. Diachronic newspaper corpus published by staatsbibliothek zu berlin., 2018.

[22] Neues Deutschland. Diachronic newspaper corpus published by staatsbibliothek zu berlin., 2018.

[23] Dominik Schlechtweg, Anna Hätty, Marco Del Tredici, and Sabine Schulte im Walde. A wind of change: Detecting and evaluating lexical semantic change across times and domains. *arXiv preprint arXiv:1906.02979*, 2019.

[24] Martin Pömsl and Roman Lyapin. Circe at semeval-2020 task 1: Ensembling context-free and context-dependent word representations. *arXiv preprint arXiv:2005.06602*, 2020.

Appendix

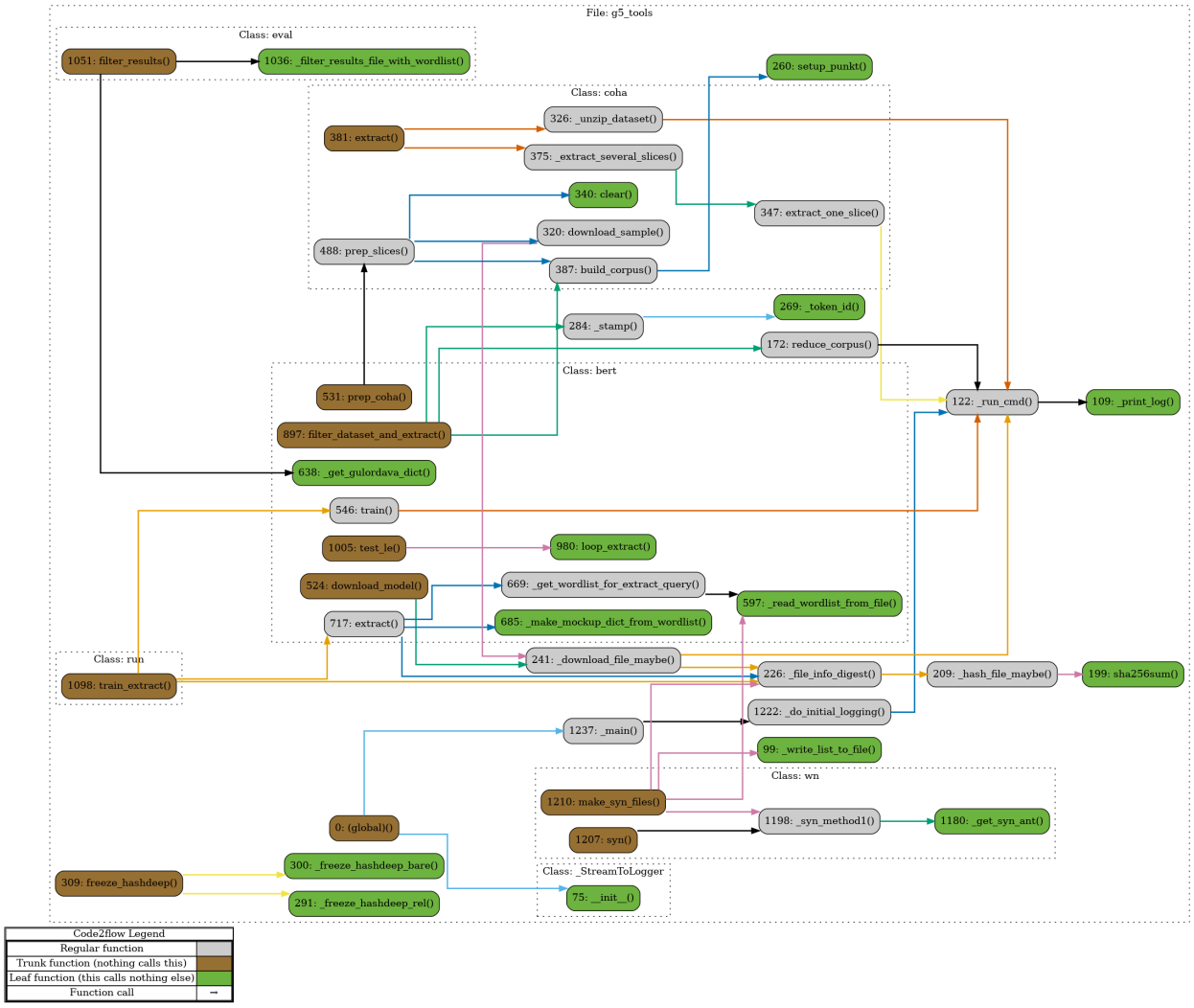

Figure 3: code2flow.g5_tools.png

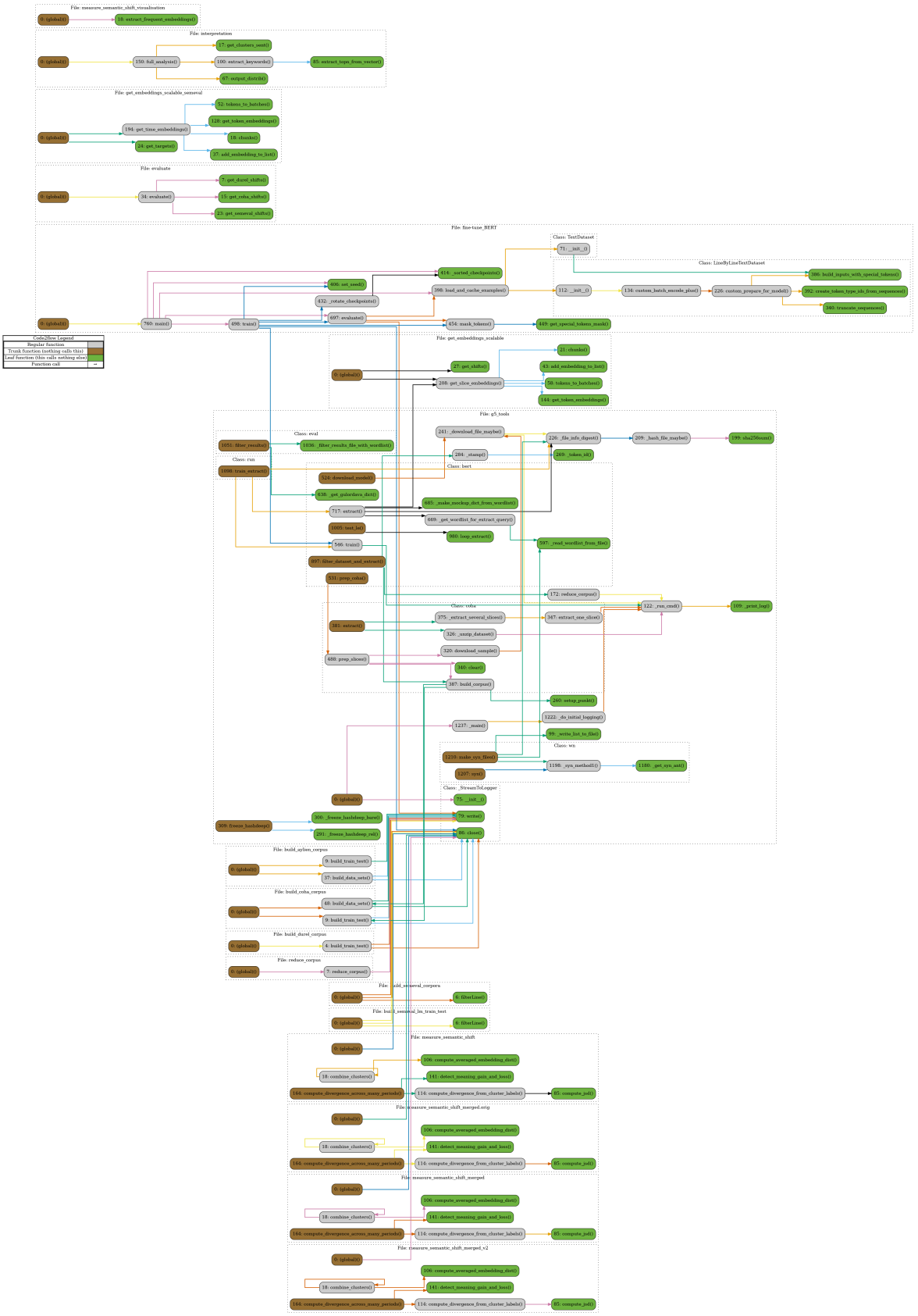

Figure 4: code2flow.repo.png (scalable semantic shift repo)

Word	Time slice	Distance from the target word
cheery	1810-1860	0.2380562424659729
low-spirited	1810-1860	0.250909686088562
pinkish	1810-1860	0.2666267156600952
festal	1810-1860	0.28348320722579956
gentle	1810-1860	0.29478001594543457
aristocratic	1810-1860	0.29766154289245605
aristocratical	1810-1860	0.30802297592163086
racy	1810-1860	0.3132709860801697
queer	1810-1860	0.3158835172653198
merry	1810-1860	0.3227323293685913
homosexual	1960-2010	0.15035313367843628
racy	1960-2010	0.23442083597183228
womanhood	1960-2010	0.2532657980918884
pinkish	1960-2010	0.2713753581047058
pink	1960-2010	0.2846895456314087
jolly	1960-2010	0.2869220972061157
queer	1960-2010	0.31636178493499756
merry	1960-2010	0.3165733218193054
bluish	1960-2010	0.31706249713897705
gentleman	1960-2010	0.3187858462333679

Figure 5: Table visualization of the top 10 closest words of the target word "gay" from the COHA corpus

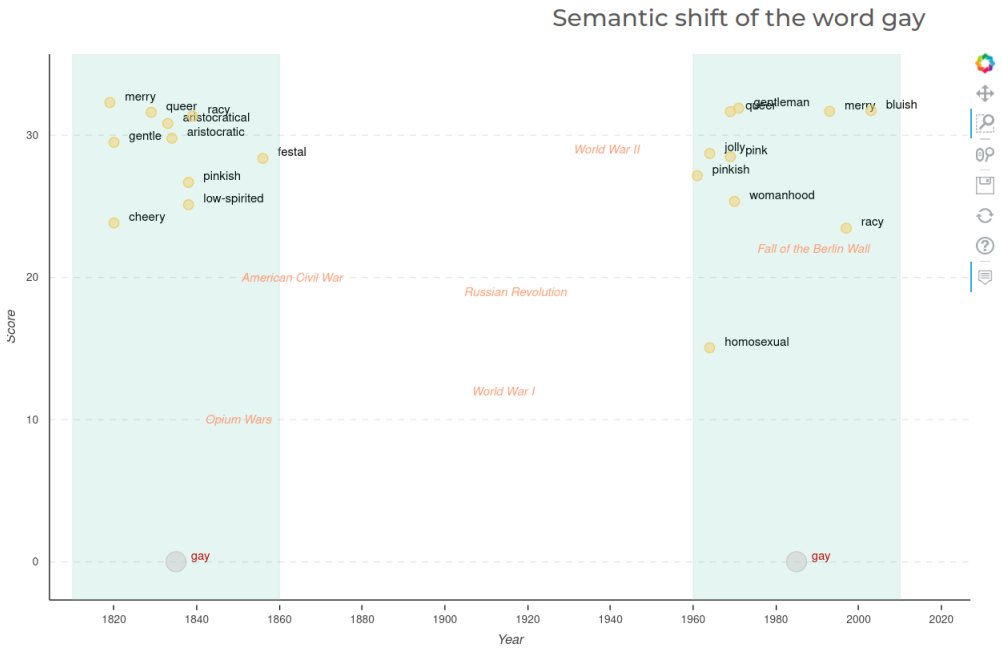

Figure 6: Cluster visualization of the word "gay" from the COHA corpus

Word	Time slice	Distance from the target word
girl	1810-1860	0.32012683153152466
aristocratic	1810-1860	0.32224780321121216
gay	1810-1860	0.32567697763442993
guy	1810-1860	0.3339216113090515
merry	1810-1860	0.3411899209022522
miss	1810-1860	0.34362077713012695
queer	1810-1860	0.3497040867805481
woman	1810-1860	0.3502378463745117
cheery	1810-1860	0.35671377182006836
father	1810-1860	0.37492620944976807
gay	1960-2010	0.2667779326438904
jolly	1960-2010	0.28007227182388306
merry	1960-2010	0.2862558960914612
homosexual	1960-2010	0.30548036098480225
cheery	1960-2010	0.3157767653465271
gentleman	1960-2010	0.3322511911392212
happy	1960-2010	0.3445558547973633
queer	1960-2010	0.3617919087409973
aristocratic	1960-2010	0.37370073795318604
girl	1960-2010	0.3809671998023987

Figure 7: Table visualization of the top 10 closest words of the target word "pink" from the COHA corpus

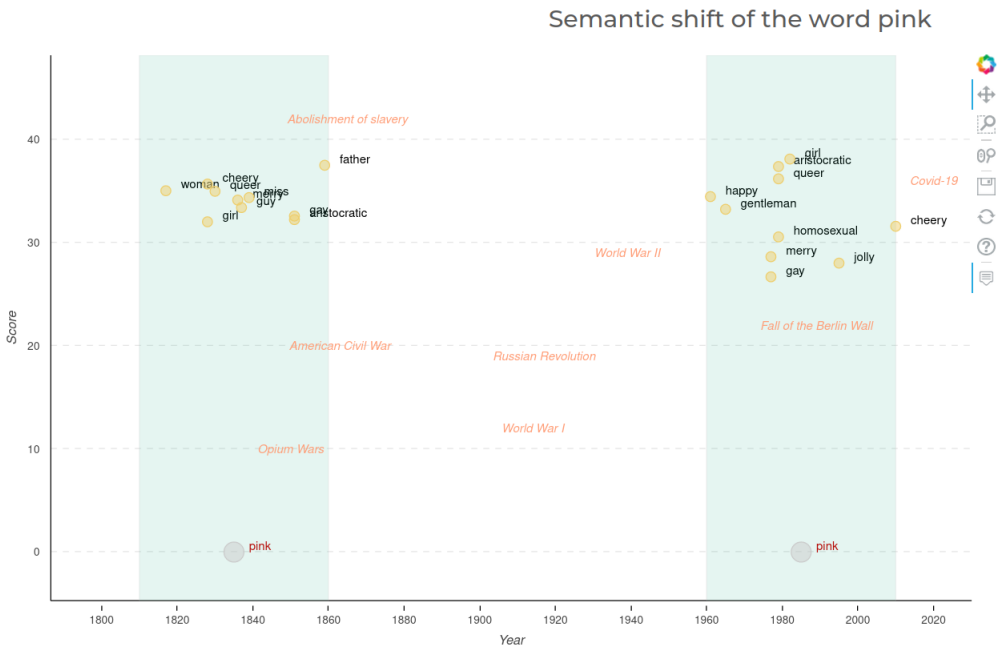

Figure 8: Cluster visualization of the word "pink" from the COHA corpus

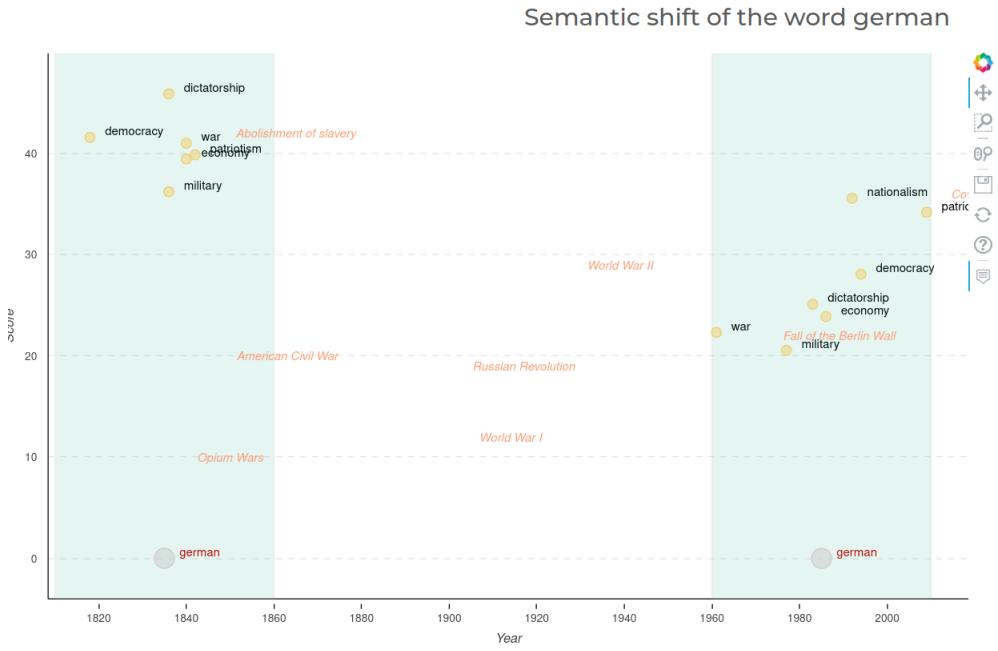

Figure 9: Cluster visualization of the word "german" from the COHA corpus

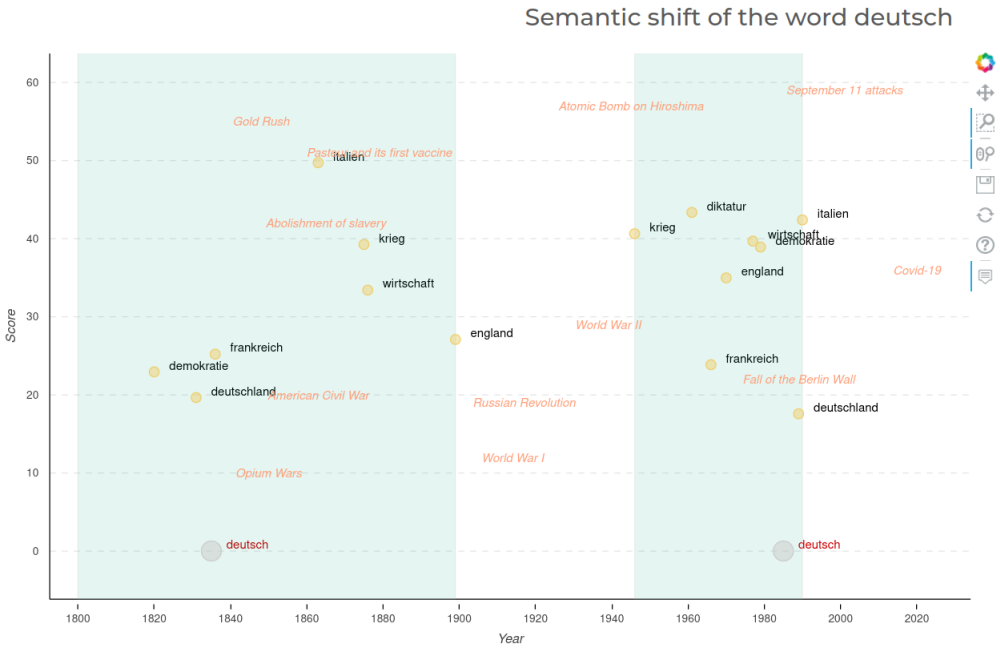

Figure 10: Cluster visualization of the word "deutsch" from the DTA corpus