OpenSemShift: Tracing and visualizing multilingual diachronic semantic change with contextualized embeddings

Software project, group 5

Averie (Ho Zoen) SO, NGO Van Duy, Scott TANKARD, Mathilde AGUIAR

Université de Lorraine

Project completed for the Software Project course (2nd-year NLP Masters) taught by Miguel Couceiro and Esteban Marquer

Tuesday 7 Feb 2023

Outline

Introduction

Motivation: why do we care?

- People have been trying to understand and record the change of word meanings for a long time e.g. *historical linguistics*
- Useful findings to many fields, beyond linguistics, e.g. history, political science, legislation – or just for fun
- Computationally, word embeddings could be a new way to represent these changes at a more representative scale

Figure 1: Example of a Google Books Ngram Viewer meme

Context: current state of research

- SemEval2020 Task 1 – Unsupervised Lexical Semantic Change Detection [1] encouraged related research
  - provided human-annotated, multilingual gold datasets
  - finding: contextualised embeddings perform worse than static embeddings - *a rare exception to the rule*

Team	Subtask 2 Avg.	EN	DE	LA	SV	System
UG_Student_Intern	.527	.422	.725	.412	.547	type
Jiaxin & Jinan	.518	.325	.717	.440	.588	type
cs2020	.503	.375	.702	.399	.536	type
UWB	.481	.367	.697	.254	.604	type
Discovery_Team	.442	.361	.603	.460	.343	ens.
RPI-Trust	.427	.228	.520	.462	.498	type
Skurt	.374	.209	.656	.399	.234	token
IMS	.372	.301	.659	.098	.432	type
UiO-UvA	.370	.136	.695	.370	.278	token
Entity	.352	.250	.499	.303	.357	type

Figure 2: Top performing systems in SemEval2020 task 1

- Nonetheless, many reasons to adopt contextualised embeddings for this task:
  - pre-trained on a large corpus, shown to be much better at most other NLP tasks
  - can capture multiple senses
  - multilingual models available
  - post-2020, some promising improvements in using contextualised embeddings for this task

Summary: so what are we trying to do?

- Beyond change detection: a visualisation tool for discovery and validation
- Intuitive fine-tuning procedure: one model per time slice
- Make it multilingual

Our Approach

Montariol et al. (2021): Scalable and Interpretable Semantic Change Detection

- Montariol et al.'s method [2]:
  1. fine-tuning BERT for domain adaptation (i.e. the entire corpus, not separated by time)
  2. retrieve contextualised embeddings for the corpus at each time slice from one single model
  3. at each time slice, cluster the embeddings for each word
  4. compare these clusters by measuring the change in the probability distribution, via Wasserstein Distance
- What we changed:
  - time-adapted multilingual models: instead of one single model, we fine-tune one model for each time slice
    - adapts the vector space to a specific period of time, so the neighbour words are also changing in each time-specific model
    - potential for application of > 2 time slices

Related Work

Semantic Change Modelling

- Semantic change is a complex phenomenon; not just change *detection*.
- Static embeddings (e.g. Word2Vec) popular in this line of research.
  - Rosin et al. [3]: time-aware search system using word relatedness (not similarity).
  - Szymanski [4]: compared meaning change over time with temporal analogies (*"Ronald Reagan" in 1987 is like "Bill Clinton" in 1997*).
- Contextualised word embeddings. Procedure of fine-tuning BERT for many NLP tasks $\rightarrow$ led to another wave of research
  - Qiu and Xu [5]: Fine-tuned BERT on COHA $\rightarrow$ more *historically balanced* model, improvements on (human-judged) temporal semantic similarity task.
  - Hombaiah et al. [6]: trained BERT incrementally for each year of tweets. Beneficial for offensive tweet classification and Country Hashtag prediction.

LSCD: Lexical Semantic Change Detection

- Static embeddings
  - HistWords [7]: tools with focus on change over historical time. Their visualisations and statistical laws of semantic change $\rightarrow$ influence on our project.
  - Arefyev and Zhikov [8]: word sense induction (WSI) via lexical substitution and clustering. Visualisations presenting clusters of word meanings and change over time.
- Contextualised embeddings
  - They outperform static word embeddings in general, but, unique challenges for detecting lexical semantic change. Multiple senses: problem but also benefit.
  - Giulianelli et al. [9]: track different senses *across* time slices. Clustering to tag the individual senses; then compare how vectors changed across original *vs* time-specific BERT models.

Methods

Data

- Train and test sets from the SemEval baseline - Task 1
- Data in 2 languages: EN and DE
- Slices: 1800s and 1900s
- Corpora:
  - EN: CCOHA$_{1,2}$
  - DE: DTA$_1$ and BZ&ND$_2$

	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

Figure 3: Corpora stats

- SemEval test set characteristics

Pipeline overview

The conceptual steps of the pipeline

- Corpus Preparation
- Fine-tuning BERT Model
- Embeddings Extraction
- Measuring Semantic Shift
  (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.)
- Afterwards: Evaluation and/or visualisation

Corpus preparation

Multilingual, time-specific corpora.

- When training on the 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. (German SemEval data is $\sim$10x larger than for English.)
- Moderate preprocessing (SemEval dataset is already preprocessed). We use the tokenized datasets from within the SemEval archive (`token/`), not the lemmatized ones (`lemma/`).

SemEval dataset is already time-sliced, but our pipeline is compatible with other time-slicing:

- Implemented slicing of the COHA sample corpus into per-year time slices. (Codepath currently for testing; can be adapted to other, larger corpora, for use in production.)
- Preprocessing step is a fully modular component within the pipeline: can integrate other languages and corpora of various types, with more fine-grained time-period granularity, in future work.

Fine-tuning BERT model

Fine-tune (unsupervised; MLM task) mBERT on time-specific corpora.

- Specifics:
  - Starting from HuggingFace's `bert-base-multilingual-cased`
  - Each model trained for 5 epochs on a GPU
  - Batch size: both 7 and 4 (...GPU RAM limitations per machine)
- Two experimental variations:
  - 1st variation: train only on English data;
  - 2nd variation: train on a concatenated file of both German and English data.
  - This results in 4 models (2 pairs of C_1 and C_2)

Scalable, flexible, generalizable approach for future use.

- Our framework can be adapted to further languages, time periods/granularity, different corpora...
- For example, narrower time-slicing (e.g. 1-year granularity) will allow to tackle large datasets in pieces (an array of models, training on a reasonably-sized quantity of data at a time).

Embeddings extraction

We extract contextualised embeddings *for* query words *from* the fine-tuned models, following the methods in Montariol et al. [2].

- We obtain embeddings for each target word based on its occurrences in the corpus (full dataset: train+test).
  - If a vector is too similar to a prior extracted one, they are merged.
  - For scalability, the threshold of vectors collected is set at 200.
- Output: a list of embeddings (each in the form of a `numpy` array) for each query word, and references to where they appear in the source corpus. $\rightarrow$ saved into `pickle` file.
- Additional prep step to reduce the time spent on extraction:
  - Filter to only keep the sentences containing at least one word stem matching list of target words.
  - Used only on querying and extracting of word-specific embeddings from the finished model. The models are trained on a downsampled but non-filtered dataset (thus: not biased toward changes of query words).
  - Benefit of large dataset for training; avoid downsides of large data when not necessary. (Gains in time; less use of computing power.)

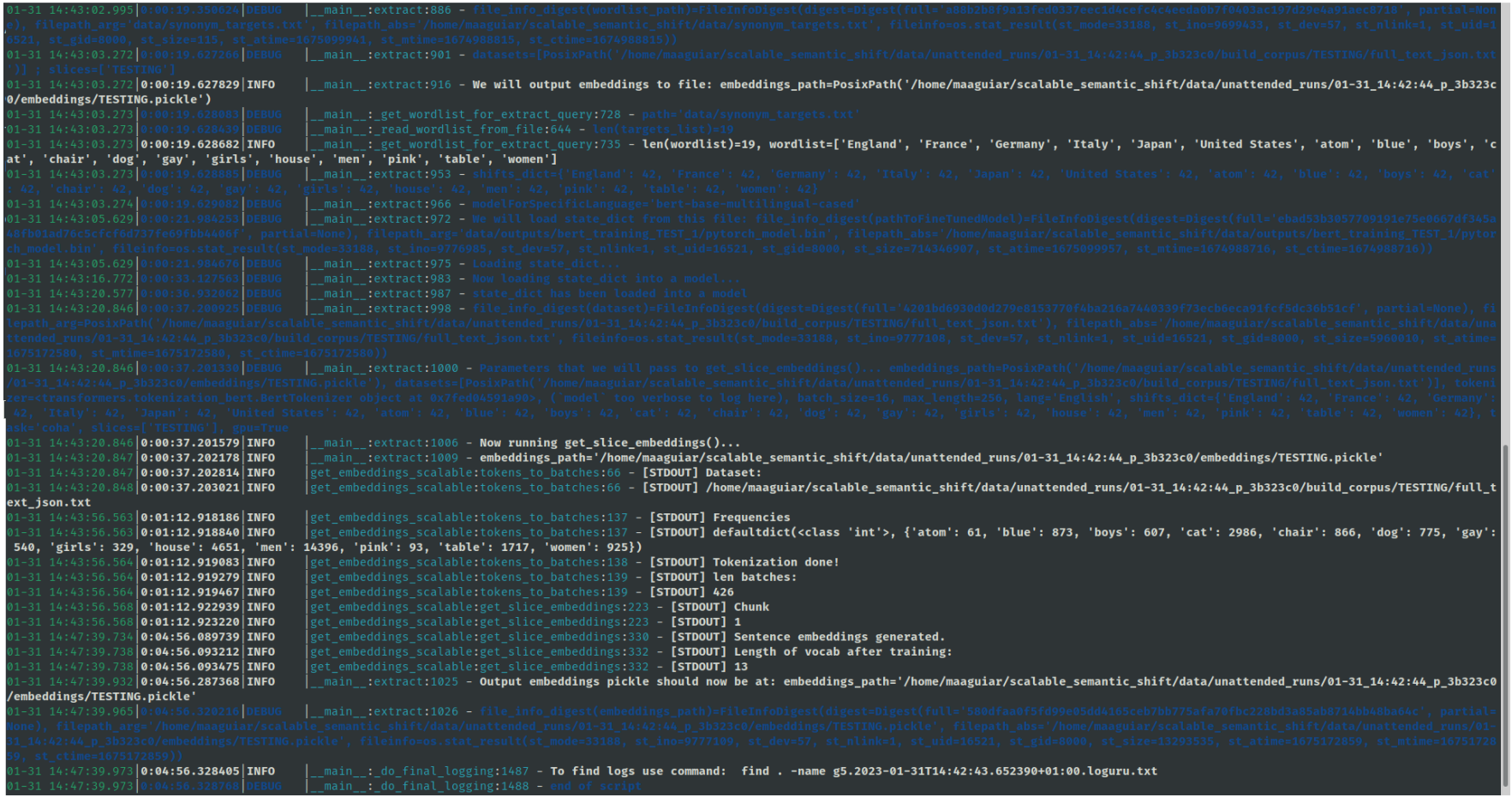

Figure 4: Screenshot of running filter_dataset_and_extract operation using g5_tools.py

Semantic Shift Measurement

`Experimental Semantic Shift:`

- Merge the embeddings from different time slices
- Cluster the usage distributions using kMeans and Affinity Propagation
- Measure the differences between the usage distributions
- Detect the most changed sense of each word
  Metrics: Jensen-Shannon Divergence and Wasserstein Distance
- Results exported to CSV files

Semantic Shift Measurement

`Visualisable Shift`:

- Retrieve the embeddings from different time slices
- Retrieve embeddings for the most frequent sense of word; compute the distances from that word to the rest of the vocabulary.
- Sort the distances and retrieve the *top n* closest neighbours
- Results exported to CSV for visualising on the web application

Evaluation

- Use the output from the `Experimental Semantic Shift`
- Retrieve the scores of semantic change from the two models
- Compute the outputs against the provided human-annotated test set
- Get the Spearman's rank correlation score

Results

Automatic Evaluation

Model	**Embeddings type**	**English**
Schlechtweg et al. (2019) [10]	SGNS	0.321
Pomsl and Lyapin (2020) [11]	SGNS	0.422
Montariol et al. (2021) [2]	BERT	**0.456**
Our model (EN), k-means 5	mBERT	0.408
Our model (EN + DE), k-means 5	mBERT	0.421

Table 1: Spearman's rank correlation with human-annotated semantic change, based on 37 target words for English. SGNS = Skip-Gram with Negative Sampling

Model	**k-means 5**	**k-means 7**	**AP**
Montariol et al. (2021)	0.360	-	**0.456**
our model (En only)	0.408	0.384	0.354
our model (En + De)	0.421	0.383	0.340

Table 2: Spearman correlation with WD as measurement, across three different methods of clustering. AP = Affinity Propagation

User Interface

OpenSemShift web app

Goal: Make our results accessible and have visualizations
Tech stack: Python Flask, Bootstrap Flask and Bokeh

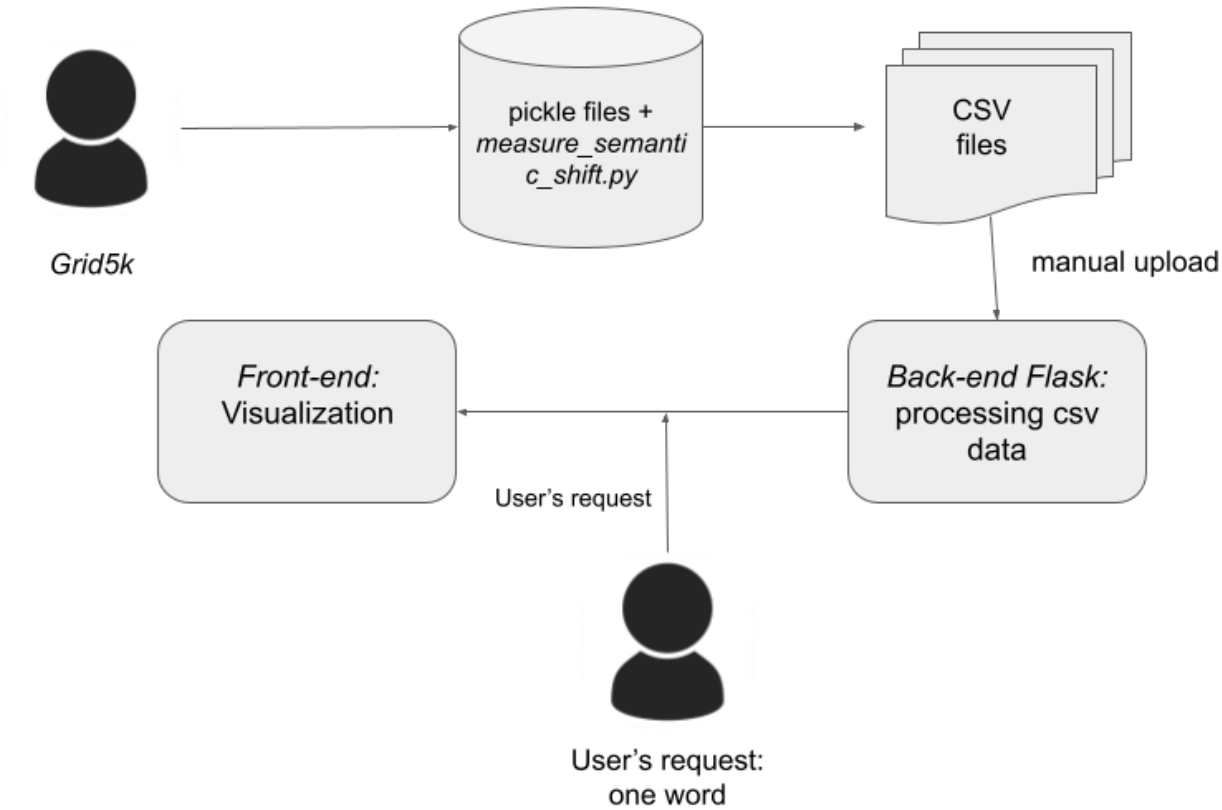

Figure 5: Overall application workflow

OpenSemShift web app

Trade-offs, reproducibility and modularity:

- Still a limited number of words available to query
- But easy to expand: generate new outputs of *measure_semantic_shift.py* and export them into the web app back-end

Case Studies

Case studies - “gay”

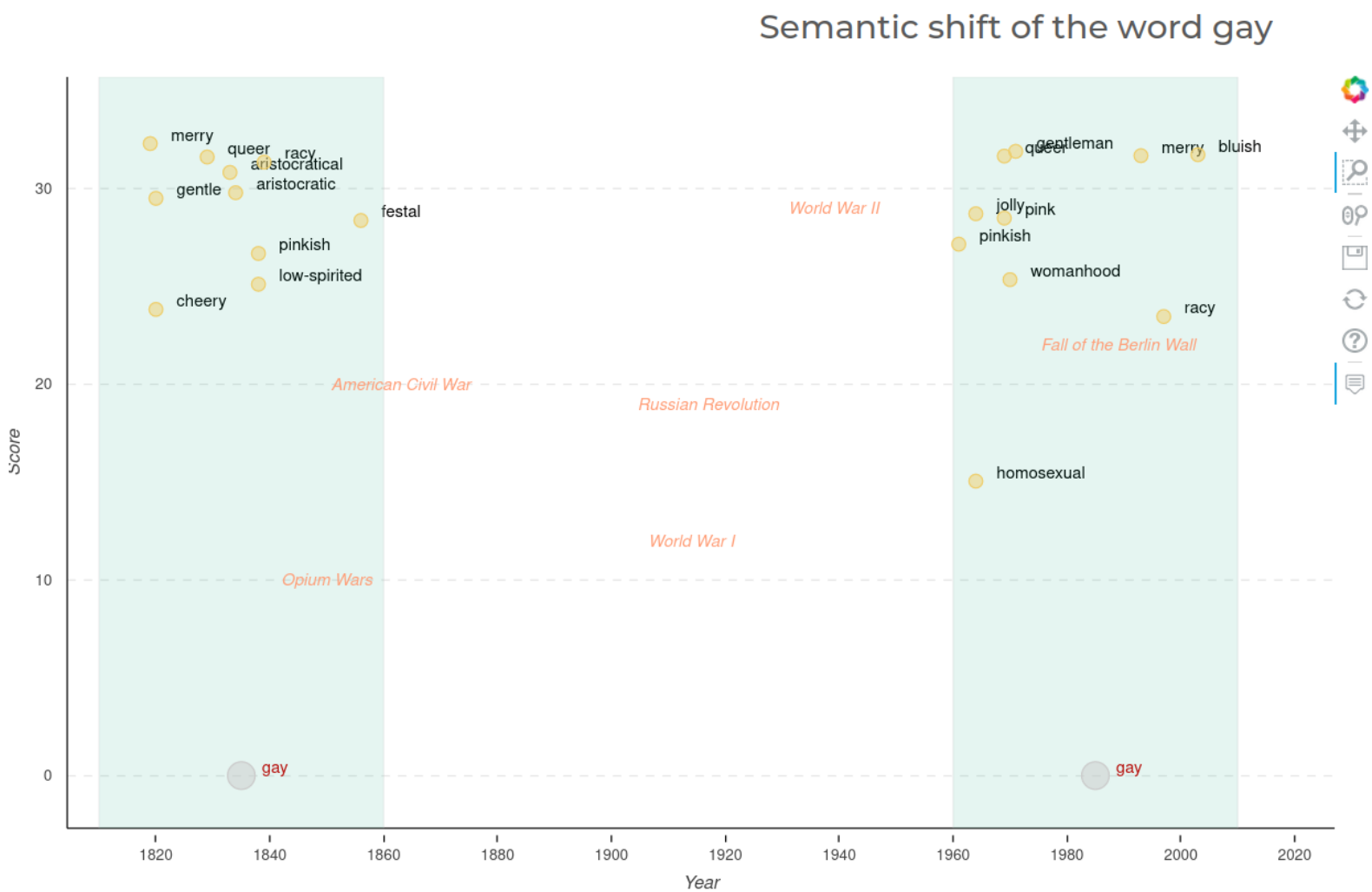

Figure 6: Scatter plot visualization of the word “gay” from COHA dataset

Case studies - “pink”

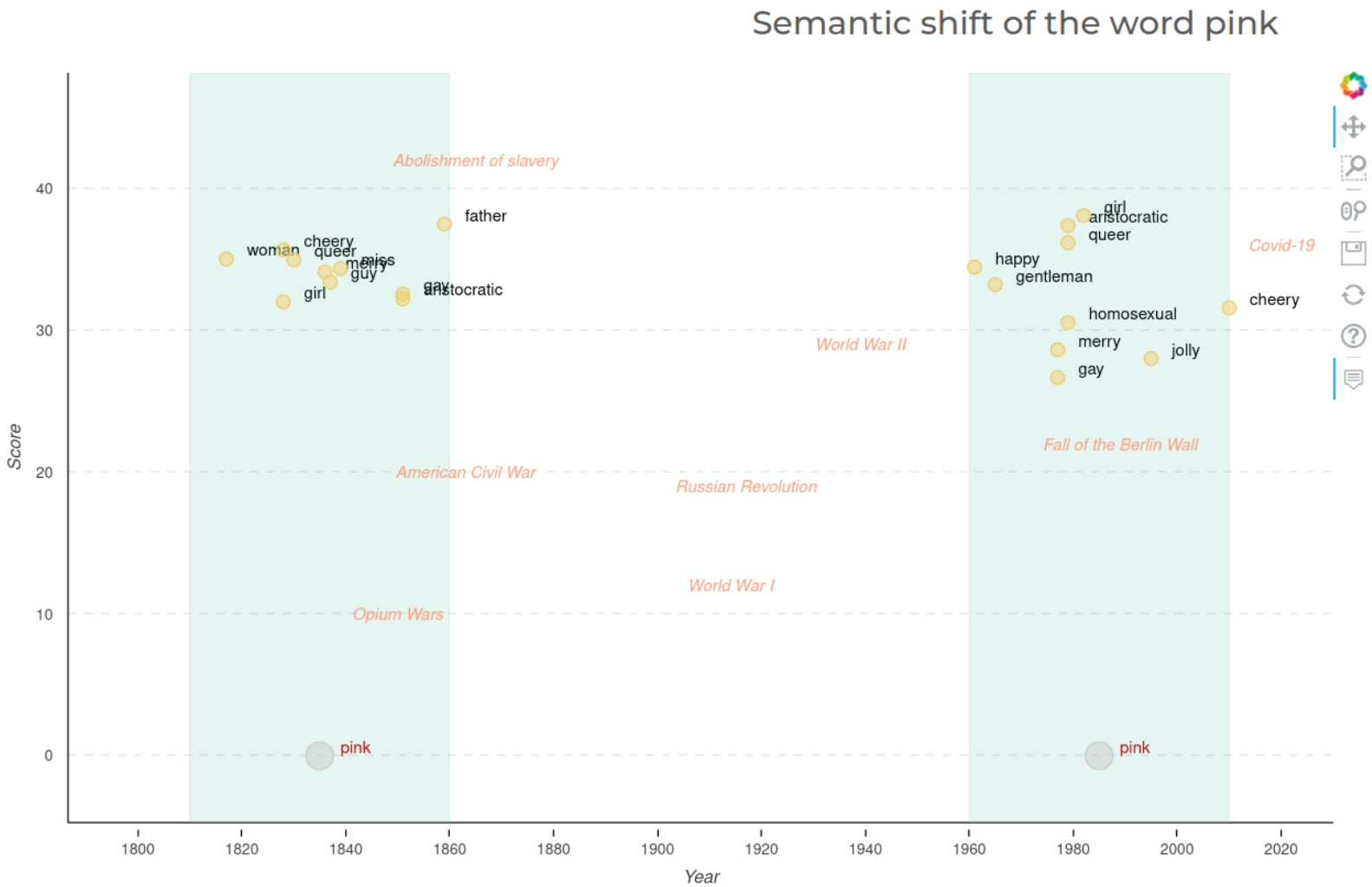

Figure 7: Scatter plot visualization of the word “pink” from COHA dataset

Case studies - “deutsch”

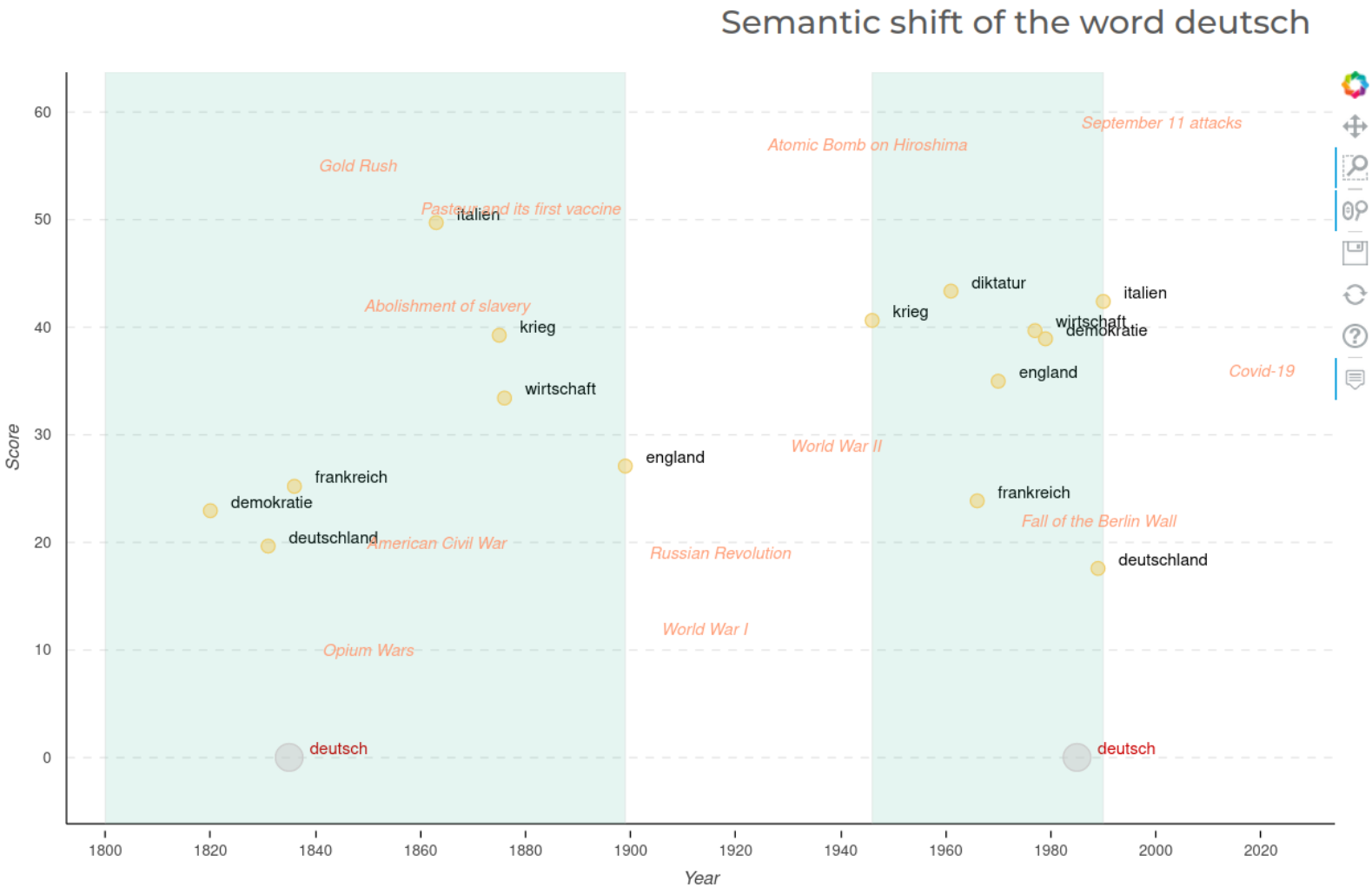

Figure 8: Scatter plot visualization of the word “deutsch” from DTA dataset

Case studies - "German"

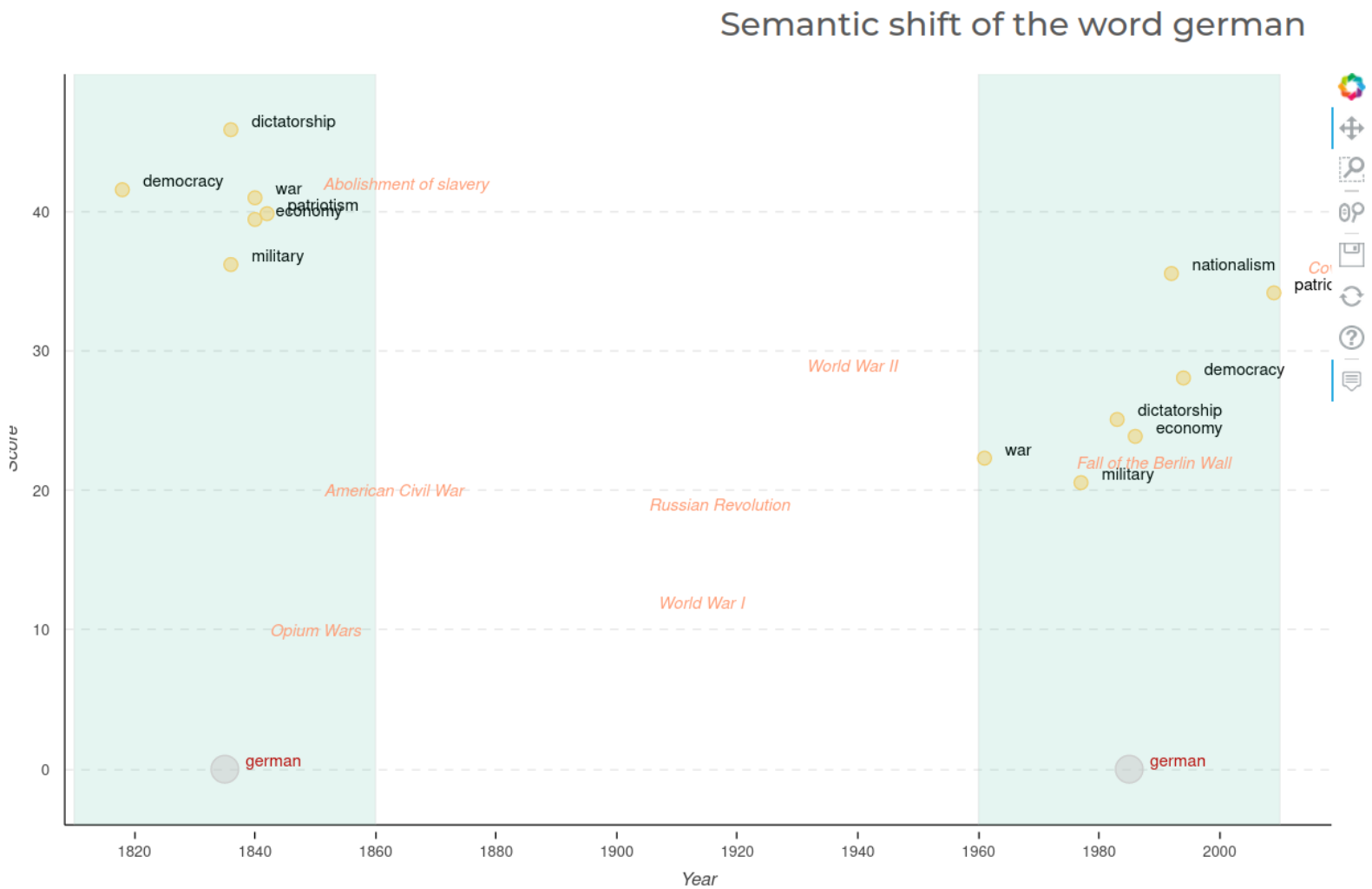

Figure 9: Scatter plot visualization of the word "German" from COHA dataset

Discussion

Discussion – Main Findings

- Summary
  - OpenSemShift is a multilingual approach to modelling lexical semantic change along with a tool that can represent semantic change visually.
  - This is achieved by fine-tuning mBERT on multiple EN and DE, time-specific corpora
  - The results are comparable to other approaches in graded lexical semantic change detection
  - Multilingually trained model has improved the performance slightly
- Analysis
  - We have not evaluated on the German test set to confirm the benefits of multilingual fine-tuning
  - Some challenges with practical concerns of deploying the software (e.g. cannot query words on the fly)
  - We have not perfected multilingual usage of the software (e.g. setting language-specific parameters)

Discussion – Outlook

- Directions for future work:
  - Extend usage to more than two time periods
  - For even more ancient texts, develop procedures for orthographic differences, as some orthographies have changed through time (e.g. English)
  - Multiple-senses *vs* single-averaged sense
  - Deepening and enriching the historical event contextualization data that we currently present in the UI
- Conclusion
  - To our knowledge, this is the first attempt in modelling semantic change with a multilingual model.
  - Our approach is competitive compared to other monolingual studies
  - Additionally, it allows for the visualisation of change among target words and their neighbours

References I

[1] Dominik Schlechtweg, Barbara McGillivray, Simon Hengchen, Haim Dubossarsky, and Nina Tahmasebi. SemEval-2020 task 1: Unsupervised lexical semantic change detection. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 1–23, Barcelona (online), December 2020. International Committee for Computational Linguistics.

[2] 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.

[3] Guy D. Rosin, Eytan Adar, and Kira Radinsky. Learning word relatedness over time. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1168–1178, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.

References II

[4] 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)*, pages 448–453, Vancouver, Canada, July 2017. Association for Computational Linguistics.

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

[6] 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.

References III

[7] William L. Hamilton, Jure Leskovec, and Dan Jurafsky. Diachronic word embeddings reveal statistical laws of semantic change. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1489–1501, Berlin, Germany, August 2016. Association for Computational Linguistics.

[8] 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.

[9] Mario Giulianelli, Marco Del Tredici, and Raquel Fernández. Analysing lexical semantic change with contextualised word representations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3960–3973, Online, July 2020. Association for Computational Linguistics.

References IV

[10] 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. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 732–746, Florence, Italy, July 2019. Association for Computational Linguistics.

[11] Martin Pömsl and Roman Lyapin. CIRCE at SemEval-2020 task 1: Ensembling context-free and context-dependent word representations. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 180–186, Barcelona (online), December 2020. International Committee for Computational Linguistics.

Many thanks to Montariol et al. [2]
for their excellent `scalable_semantic_shift` codebase.

Question time