When Time Makes Sense: A Historically-Aware Approach to Targeted Sense Disambiguation

Kaspar Beelen[*]
The Alan Turing Institute
Queen Mary University of London
kbeelen@turing.ac.uk

Federico Nanni
The Alan Turing Institute
fnanni@turing.ac.uk

Mariona Coll Ardanuy
The Alan Turing Institute
Queen Mary University of London
mcollardanuy@turing.ac.uk

Kasra Hosseini
The Alan Turing Institute
khosseini@turing.ac.uk

Giorgia Tolfo
The British Library
giorgia.tolfo@bl.uk

Barbara McGillivray
The Alan Turing Institute
University of Cambridge
bmcgillivray@turing.ac.uk

Abstract

As languages evolve historically, making computational approaches sensitive to time can improve performance on specific tasks. In this work, we assess whether applying historical language models and time-aware methods help with determining the correct sense of polysemous words. We outline the task of *time-sensitive Targeted Sense Disambiguation* (TSD), which aims to detect instances of a sense or set of related senses in historical and time-stamped texts, and address two main goals: *1)* we scrutinize the effect of applying historical language models on the performance of several TSD methods and *2)* we assess different disambiguation methods that take into account the year in which a text was produced. We train historical BERT models on a corpus of nineteenth-century English books and draw on the Oxford English Dictionary (and its Historical Thesaurus) to create historically evolving sense representations. Our results show that using historical language models consistently improves performance whereas time-sensitive disambiguation helps especially with older documents.

1 Introduction

As language is in continuous flux, the question arises as to whether (and how) we should adapt Natural Language Processing (NLP) methods to the changing context in which texts are produced.[1] This paper offers a novel contribution by assessing the extent to which making NLP models sensitive to time (or historically-aware) actually improves their performance. We present the task of time-sensitive *Targeted Sense Disambiguation* (TSD), which determines whether or not a token in a given time-stamped text is related to a specific sense of a lemma. TSD is a variation on Word Sense Disambiguation (WSD), but (as we argue) it is of more practical relevance to research in digital history and cultural analysis. For example, if a historian wants to investigate the lemma *machine*[2] in the sense of "anything that transmits force or directs its application", TSD classifies whether the token *machine* (or one of its synonyms, e.g. *machine*) in a given text expresses this sense. This task assists researchers with tracking the evolution of very specific senses across time, instead of just words, as in Michel et al. (2011).

[*] Contributions of each author (in alphabetical order): Conceptualization: KB, BMcG, FN; Data curation: KB, GT; Formal Analysis: MCA; Funding acquisition: BMcG; Methodology: KB, MCA, KH, BMcG, FN; Project management: KB, BMcG, FN; Software: KB, MCA, KH, FN; Supervision: BMcG; Reproducibility: KB, MCA, FN; Writing: KB, MCA, BMcG, FN.

[1] Code and models used in this paper are accessible via Github and Zenodo, see https://github.com/Living-with-machines/TargetedSenseDisambiguation and https://zenodo.org/record/4782245.

[2] In this paper we will refer to lemmas or tokens in italics, their senses in single quotes and full definitions in double quotes.

Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pages 2751–2761
August 1–6, 2021.

For our experiments, we use the Oxford English Dictionary (OED),[3] which provides a very detailed insight into sense-level change over time, exemplified by time-stamped quotations from a large collection of historical texts. The OED is very fine-grained, with a high number of senses for each entry. The number of historical examples for each sense is limited however, which presents a challenging data sparsity setting,[4] a not uncommon problem in WSD research and also characteristic of Digital Humanities (DH) research, in which often only a small amount of positive examples are available.

Methodologically, our approach to time-sensitive TSD builds upon recent advances in WSD and leverages BERT architectures and contextualized word vectors to address two questions:

RQ1. (time-sensitive embeddings) *Do BERT language models, fine-tuned on specific epochs, yield better representations for historical TSD?* In this scenario, we assess if models trained on data from a certain year range work better on contemporaneous texts compared to standard BERT models.

RQ2. (time-sensitive disambiguation) *When confronted with scarce historical examples, do time-sensitive methods outperform those which ignore the temporal context?* These experiments focus on the impact of time-sensitive strategies.

2 Related work

The main point of reference for Targeted Sense Disambiguation (TSD) is the extensive literature on Word Sense Disambiguation (WSD). WSD tasks can be divided into *lexical sample* and *all words* tasks (Navigli, 2009): in the former approach, models disambiguate a specific (polysemous) word in context, whereas the latter disambiguates all words in a sentence.

WSD research has a long tradition and has achieved good results in synchronic settings (Navigli, 2009; Bhattacharjee et al., 2020). However, time-sensitive WSD has received very little attention, with most literature reviews (e.g. Navigli (2009), Ranjan Pal and Saha (2015), Aliwy and Taher (2019), and Bhattacharjee et al. (2020)) not including it. In spite of the little attention received, time-sensitive WSD has important applications to DH research and the cultural heritage sector. For many applications, it is important to mine historical texts semantically, especially for historical information retrieval, OCR correction and broader research areas such as cultural analytics, as surveyed by (Tahmasebi et al., 2018, 46-47).

As far as we know, Piao et al. (2017) is the only work focusing on a time-sensitive all-words WSD system. The authors present the Historical-Thesaurus-based Semantic Tagger (HTST), a tool to annotate all lexical units of texts with the semantic categories from the Historical Thesaurus of English (Kay et al., 2016). The method by Piao et al. (2017) does not make use of corpus-driven models of word semantics. Over the past few years, a growing body of research has focused on this aspect, and researchers have developed different models for representing words' changing meaning over time. These studies have traditionally employed word embeddings models (Tahmasebi et al., 2018; Kutuzov et al., 2018), which conflate the different senses of words into a single representation. Some work has modelled the diachronic distribution of word senses (Mitra et al., 2014; Tahmasebi and Risse, 2017).

Hu et al. (2019) report on a method for building sense representations using the mapping between example sentences and sense definitions from the synchronic data of the Oxford Dictionary of English.[5] They focus on 4881 target words chosen based on a frequency filter on the COHA corpus. They first feed up to 10 sentences for each sense of every target word to a pre-trained BERT model; they then use the target word's token embeddings from the dictionary's example sentences and average them to obtain 768-dimensional embeddings for its senses. The correct sense is assigned to a token in context by finding the sense whose embedding has the highest cosine similarity score with the token embedding. They also apply their approach to track the development of individual senses of a target word over time via time series decomposition. Their system is able to trace fine-grained lexical semantic shifts as a smooth process, obtaining an improvement over previous models by Frermann and Lapata (2016) and Gulordava and Baroni (2011).

[3] www.oed.com.

[4] At the time of writing, the OED contains over 270,000 entries, each of them associated with one or more senses for a total of 800,000 senses (so approximately 3 senses per entry, on average), most of which have one or more quotations associated to them, for a total of over 3 million quotations (so approximately 3.75 quotations per sense, on average). See https://public.oed.com/how-to-use-the-oed/glossary/ for details.

[5] www.lexico.com.

Gonen et al. (2020) propose a new approach to detect usage change of words across corpora that is more stable and interpretable, using the differences in the top nearest neighbors of a word in a vector space as a proxy for usage change of that word. More recently, methods based on token embeddings have shown competitive results (Schlechtweg et al., 2020). Giulianelli et al. (2020) propose the first method for using contextualized (BERT) word embeddings to model sense distributions over time. They build token embeddings and then cluster them into "usage types" (which can be interpreted as senses) using K-means clustering. They then build a probability distribution from the frequencies of these usage types and use it to measure lexical semantic change.

Recent years have seen an increased interest on the application of such methods for modelling semantic change to DH research, primarily using unsupervised methods (McGillivray et al., 2019; Soni et al., 2021). However, in order to be useful, analyses for DH research require a high degree of granularity on highly complex datasets and this has not yet been achieved by state-of-the-art methods. This paper proposes a method which addresses these challenges and is therefore directly relevant and applicable to DH research.

3 Task Definition

We define the task of targeted sense disambiguation as follows: given a target sense σ (realized as a lemma-sense pair), the goal is to determine whether a token τ in a context κ is relevant to the sense σ. For evaluation purposes, we measure relevance by considering only tokens whose sense is identical or synonymous of the target sense. According to this definition, we formulate the task as a *one-vs-all* classification problem, where instances (either definitions or quotations) of the relevant sense (or senses) are considered as examples of the positive class and instances of the remaining senses are regarded as examples of the negative class.

Example. The OED lists twenty-six senses for the lemma *machine*. In this example, we will consider that the relevant sense is the one corresponding to the following definition: "A complex device, consisting of a number of interrelated parts, each having a definite function, together applying, using, or generating mechanical or (later) electrical power to perform a certain kind of work". Example 1 shows a positive (class 1) instance of this sense in a quotation containing *machine*; example 2 is a negative instance of this sense (class 0); example 3 is a quotation that shows a positive instance of a synonymous sense 'plant' (as in mechanical plant, therefore class 1); quotation 4 is a negative example of 'plant' (class 0).

(1) *The calculating **machine** now constructing under the superintendence of the inventor.*

(2) *The Church was excellent as a national refrigerating **machine**.*

(3) *Examples of mobile earthmoving **plant** are bulldozers, graders and scrapers.*

(4) *I could lift the **plant** and be far away before daylight.*

In contrast to the closely-related task of WSD, the objective of TSD is not to provide the most relevant sense for each attestation of a lemma, but to discriminate whether or not a token in a context is related to the pre-selected sense(s). Its aim is to find occurrences of senses related to a selected set of query senses (with the "relation" being an adjustable parameter in the hands of the user, in our case this is synonymy). Importantly, our setup is not static and depends on the user input: we derive an extended group of senses from an initially selected set of relevant sense(s), time period and relation(s). Therefore, depending on these decisions, each of the retrieved senses could either be a positive or a negative example, which makes the task substantively different. Lastly, in our flexible one-vs-all approach, the positive class may be realized by a group of senses (for instance senses that share a common characteristic relevant for the downstream research task) and is suitable to many text mining and information retrieval applications in DH.

4 Data

An important motivation for this work is to provide an efficient framework that leverages semantic information encoded in historical dictionaries and thesauri for research on time-stamped texts.

The Oxford English Dictionary. Considered "the definitive record of the English language" on its webpage,[6] the OED is the result of decades of careful curation by lexicographers. Senses are exemplified by quotations collected over time from

[6] https://www.oed.com/

different types of sources, mostly literary works, newspapers, journals, and other periodicals.[7] Each quotation has the form of a time-stamped text snippet containing the headword of the entry, the sense it represents, and is provided with metadata such as the author and title of the source. Each sense is associated with a definition, and is provided with metadata such as the date range of use and the date of its first occurrence. The example below shows a quotation for the noun *machine* with relevant metadata:

Headword: *machine*, n.
Quotation: *Windmills as hitherto made are very costly machines.*
Sense ID: machine_nn01-38475286
Definition: *"A complex device, consisting of a number of interrelated parts, each having a definite function [...]"*
Text daterange: 1659-
Keyword: *machines*
Offset: 43
Year: 1881
Source: *Nature*, by W. Thomson.

Senses are also linked to semantic classes in the Historical Thesaurus of English (HTE), therefore providing access to synonyms and other semantically related senses.

Data preparation. Although our methodology is generally applicable to any word, in our experiments we focused on a set of twelve headwords, selected because these are complex notions, spanning from political terms like *nation* to gendered and emotion words, or ambiguous terms such as *apple*.[8] Each selected headword has multiple senses.

To thoroughly evaluate our approach, we considered as many experimental scenarios as there are senses for a given headword. In each scenario, one of the headword's senses is the targeted sense (also called "seed sense"): its quotations are labeled as positive instances, while the quotations of the remaining senses are labeled as negative instances. In addition, we used the relation between the OED and the HTE to retrieve synonyms for each sense.[9] Henceforth, we will use the terms **seed sense** and **synonym sense** to distinguish between them. We thereupon expanded the set of positive and negative instances as follows:

Positive class. All quotations pertaining to a synonym of the targeted seed sense were labeled with the positive class. This means that we enriched the set of positive examples with quotations that have a different lemma than the original headword (e.g. the mechanical *plant* in the example above).

Negative class. The new lemmas (included through synonymy to expand the set of positive examples) are often ambiguous: they may refer to senses that bear no relation to the seed sense (henceforth **unrelated senses**). Therefore, for each of the new lemmas, we collect the quotations of the unrelated senses as well, and label them with the negative class.

This data expansion step allows us to overcome the problem of data sparsity, since the number of quotations per sense is generally quite low (3.75 on average). In each data set, we removed unambiguous words from the set of expanded lemmas, and held out 25% of the quotations for testing, which is consistently the same for all methods and baselines (whether supervised or unsupervised). For supervised baselines and methods, the remaining 75% is further split into training set (80%) and validation set (20%). Baselines and methods that do not require the distinction between training and validation use both as one. Finally, we consider time to be a determining factor. Therefore, our train, validation and test sets are filtered by time.

Table 1 lists the headwords with the number of seed senses, the averaged number of synonyms and unrelated senses for each seed sense (expanded senses), and the averaged number of the positive and negative quotations per seed sense for the period between 1760 and 1850.[10]

5 Experimental Design

To measure the impact of time, we designed two experiments, each addressing a research question:

[7] See https://oed.hertford.ox.ac.uk/ for an in-depth examination and analysis of the contents and sources of the OED.

[8] We focused on the following nouns: *anger*, *apple*, *art*, *democracy*, *happiness*, *labour*, *machine*, *man*, *nation*, *power*, *slave*, and *woman*. We choose to avoid classical semantic change examples, such as *cell*, *gay* or *mouse*, as the semantic evolution of these words is generally well known and not that relevant to current historical or DH research. The number of selected examples was determined by the call limit of the OED API. The OED API is available for researchers upon request. We provide the code to replicate the entire data extraction and processing, as long as the user has OED API credentials.

[9] According to the OED Researcher API documentation, these "may not always be precisely synonymous [...]. Co-occurrence of senses in the same semantic class tends to mean that they are semantically very close, not necessarily synonymous": https://languages.oup.com/research/oed-researcher-api/.

[10] Since there is no space to report the exact numbers for all the experimental configurations, we show the numbers for the experiment that has the strictest filters.

	Seeds	Expanded senses	Quotations
anger	6	17/121	103/564
apple	19	6/61	36/300
art	19	5/47	33/212
democracy	7	6/61	37/301
happiness	5	9/46	57/186
labour	18	4/30	29/148
machine	25	8/83	42/361
man	48	9/78	55/380
nation	15	8/85	53/430
power	39	5/49	34/244
slave	10	20/158	103/670
woman	17	10/81	64/379

Table 1: Headwords used in the experiments, with the number of seed senses, their expanded senses (synonym and unrelated) averaged per sense, and number of quotations (positive and negative examples) averaged per sense.

Experiment 1. To measure **RQ1** (impact of language model fine-tuning), we produced two historical BERT models, trained on different subsections of a 19th century book corpus:[11] one on books predating 1850 (referred to as *BERT_1850*) and one on the whole collection (referred to as *BERT_1900*). To quantify the impact of fine-tuning, we compared the performance of historical models with a standard BERT model (*BERT base uncased*) on different time-stamped subsections of our dictionary data. For each of the three different epochs e (1760-1850, 1760-1920 and 1760-2000) we followed the data preparation procedure as explained in 4 and moreover removed quotations for senses that are not current ("alive") in e.[12] In the test set, we removed all quotations that fall outside the date range defined by e (to establish how well our models work for this specific period based on training data with historically relevant senses). The selected periods align with the different language models we trained on the 19th century corpus, with exception of the last one (1760-2000), which was included to assess if fine-tuning hurts performance when more modern data are included. We hypothesized that the language model closest to the target period will yield the highest scores.

Experiment 2. To answer **RQ2** we followed the same procedure as described above, but focused on evaluating the impact of multiple time-sensitive approaches which take both a token *and* a time stamp as input. As we are primarily interested in understanding how well TSD works as a tool for historical analysis, we only compute scores for the periods 1760-1850 and 1760-1920.

We should stress at this point that the task we pursue is hard, given the complexity of the target concept we attempt to disambiguate (the selected sense and its synonyms) and the minimal number of historical examples at our disposal. At the same time, this makes TSD an excellent task for assessing the gains of historicizing NLP methods. Moreover, we argue that (even taking into account these limitations) TSD is a pragmatic and efficient task to assist with the exploration of historical texts.

6 Embedding Models

In Section 8, two types of language models are used: BERT (contextualized word representations; Devlin et al. 2019) and word2vec (static word representations; Mikolov et al. 2013):[13]

BERT. We used the *BERT base uncased* model and tokenizer as contemporary model,[14] hereinafter referred to as *BERT_base*. To investigate the impact of time on language models, we generated two historical BERT models, *BERT_1850* and *BERT_1900*, by fine-tuning *BERT_base* on a collection of historical books in English digitized by the British Library in partnership with Microsoft (henceforth *MBL*).[15] In *BERT_1850*, the contemporary BERT model was fine-tuned on the historical books published before 1850 (with ≈1.3B words). In *BERT_1900*, all MBL books were used for fine-tuning (≈5.1B words).[16] To fine-tune these models, we firstly preprocessed all books[17] and tokenized them using the original *BERT_base* tokenizer as implemented by HuggingFace[18] (Wolf et al., 2019). The tokenized sentences were then fed into the language model fine-tuning tool in which only the masked language model (MLM) objective was optimized.[19]

[11] See Section 6 for more information.

[12] Please note that we retain the senses that *overlap* with e, which entails that some quotations will have dates outside the range of e.

[13] See Hosseini et al. (2021) for a more detailed description of the historical language models.

[14] https://github.com/google-research/bert.

[15] Available via https://doi.org/10.21250/db14 (British Library Labs, 2014).

[16] Note that this dataset includes a few books published after 1900, however, the large majority predates 1900.

[17] We converted the text to ASCII, fixed common punctuation errors, dehyphenated broken tokens, removed most punctuation and separated the remaining punctuation marks from tokens, and finally split token streams into sentences using the *syntok* library: https://pypi.org/project/syntok/.

[18] https://github.com/huggingface/transformers.

[19] The MLM probability was set to 0.15. We used a batch size of 5 per GPU and fine-tuned for 1 epoch over the books.

Word2Vec. We used all the MBL books to train the *w2v* model using skip-gram algorithm as implemented in Gensim (Řehůřek and Sojka, 2010). For training, we chose a context window of five words and embeddings of size 300.

7 Disambiguation Methods

We present here the different approaches tested in our experiments.

7.1 Sense Embeddings from Transformers

Inspired by previous work by Hu et al. (2019) and Kutuzov and Giulianelli (2020), which leveraged contextualized word embeddings built on concrete historical examples, we start by extracting BERT embeddings for each quotation keyword in the training data. The vector we obtain is the concatenation of the last four layers.[20] We then average these keyword vectors by either label or sense: the former method creates a *binary centroid* (one for each class), the latter one for each sense (*sense centroid*). For each quotation in the test set, we produce a vector for the keyword (using the same procedure). In the case of the binary centroid, we assign it to the class of the nearest neighbour; for the sense-level centroid we obtain all sense embeddings that match the lemma of the keyword, and take the class of the nearest sense-centroid, based on cosine similarity.

7.2 Diachronic Sense Embeddings from Transformers

In this scenario, we allow the model to use the timestamp of a quotation to adjust the sense centroid. We compared two broad strategies, namely *filtering* and *weighting*. The filtering approach takes into account those observations in the training data that are temporally close (in absolute distance as $abs(year_{train\_example} - year_{target})$) and ignore the rest. In our experiments, we used the keyword vector of the temporally closest quotation (henceforth *nearest*). The weighting approach (henceforth *weighted*) takes a weighted average over vectors, with the weight determined by the temporal distance to the target quotation. For each example in the training data we *a)* first compute the temporal proximity of the target quotation (from the test set) as $1/(abs(year_{train\_example} - year_{target}) + 1))$ and normalize these scores so they add up to one; and *b)* we multiply the context vector of the keyword by this normalized score. We then simply add these time-weighted keyword vectors into one time-weighted sense centroid, after which we use the same procedure as described in section 7.1.[21]

7.3 Binary Perceptron

Instead of aggregating (collapsing all vectors into binary or sense centroids, in which useful information could get lost), we added one more method that directly uses the keyword vectors extracted from BERT to train a binary classifier. In this case we used a single fully-connected neural network (perceptron) with a RELU activation function (which equates to freezing the BERT model and only fine-tuning one fully-connected layer).

7.4 Baselines

In order to better understand the performance of each method in the different evaluation settings, we compare them to a set of widely established baselines:

Random. First of all, a random baseline, to measure the overall experimental complexity.

Lesk. Then a group of baselines measuring with different strategies the similarity between the positive sense definition and the given textual context. They do so by assessing **token overlap** and **sentence embedding**[22] cosine similarity. Such baselines, in comparison with the next one, show whether it is overall better to rely on the given definitions or on a (small number) of positive examples of quotations.

Supervised classifier. We finally present a Support Vector Machine (SVM) as a supervised binary

The choice of batch size was dictated by the available GPU memory (we used 4× NVIDIA Tesla K80 GPUs in parallel). Similar to the original BERT pre-training procedure, we used the Adam optimization method (Kingma and Ba, 2014) with learning rate of 0.0001, $\beta_1 = 0.9$, $\beta_2 = 0.999$ and L_2 weight decay of 0.01. In our fine-tuning routine, we used a linear learning-rate warmup over the first 2,000 steps. A dropout probability of 0.1 was used in all layers.

[20] If a word has multiple subtokens, we averaged their vector representations.

[21] We have experimented with more complex methods, for example using a Gaussian distribution centred on the time of the target quotation to compute weights. As these methods were more complex but hardly showed any improvements, we decided to only report scores for the simpler implementations in the tables below.

[22] We generate sentence embeddings by element-wise average of their word embeddings (here we used the historical Word2Vec model), a common strategy for a well-performing baseline (Shen et al., 2018).

baseline classifier, trained on sentence embedding representations of positive and negative examples of quotations.

8 Evaluation

As mentioned above, in all cases, the experimental setting is very unbalanced, with just a few quotations as positive examples, as opposed to many negative ones. In order to assess the role that *time* plays in these experiments, we report the performance of each method in terms of precision, recall and F1 Score with respect to the *positive class*. Such evaluation highlights which method is most suited to identify occurrences of a pre-determined sense and will also clearly pin-point specific limitations of each approach (for instance methods with high precision, but low recall).

The tables below report performance averaged across all senses and words under study.[23]

Experiment 1. Table 2 shows that BERT approaches outperform the established baselines. Across all models and periods, the *sense centroid* approach returns the highest scores. Moreover, the results confirm our initial intuition that BERT works better when it is fine-tuned on data contemporaneous to the target period. We see for example how the performance of the BERT model trained on the first half of the nineteenth century decreases faster than the other models when more modern data is added, whereas conversely BERT base performs worse (compared to the other models) when older data is added. BERT models that have observed historical data perform better on our examples from the 19th century. Even when more recent data is added, these fine-tuned models work well, although the scores tend to converge. To gauge if historical models perform better, independently of the method used, we computed the gains (or losses) of plugging in different language models directly for all approaches. We observed that while fine-tuning does not always guarantee a jump in performance—e.g., in the binary centroid method the F1-Score declines—the overall improvement is stable for the *BERT_1900* model: given a wide enough range of document, the historical models produce higher scores. The base model fine-tuned on the whole MBL books collection seems to work best for all experiments.

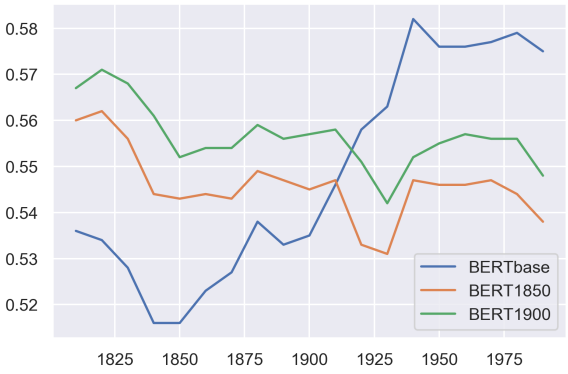

Figure 1: Optimal date range for each language model as measured by the F_1 score of the positive class, using the *sense centroid* method: the x-axis represents the average points of rolling 100-year quotation date ranges.

To understand the importance of time, we repeated the experiment above, now using a rolling time-window of 100 years (and step size of 10 years) as our historical periods. Figure 1 presents the performance of our best method in relation to the language model and date range of quotations, for all the headwords. It shows a clear jump in performance for BERT_base towards the later decades of the twentieth century, clearly surpassing the models fine-tuned on nineteenth-century data, which slowly decrease as time advances. A more in-depth analysis of the data indicates that some concepts have a stronger attachment to the period of the data than others. Setting *machine* and *power* apart from the rest of the headwords, for example, indicates that they are responsible for large part of this difference between the language models. In general, *BERT_1850* is the most suitable model for early 19th century sentences, while *BERT_base* clearly outperforms the other models on later data.

If we are to mildly speculate, and attempt to deduce more general lessons from these experiments, then fine-tuning on a historical large corpus could improve results, and is preferable when scrutinizing the semantics of heritage collections. However, success is not guaranteed, as the BERT model trained on books prior to 1850 does not yield the best results when evaluated on quotations from the first half of the century. Higher granularity does not always deliver gains in accuracy, but it does not harm either: note that *BERT_1850* still improves over the base model from which it was originally derived. Also, *BERT_1900* seems to work better

[23] All code and models for reproducing these tables are accessible on Github and Zenodo. We should warn that results may slightly differ to those reported below, because the order in which the data are retrieved (using the OED API) and the (number of) quotations themselves might vary. Data used in this paper was downloaded in December 2020.

	1760–1850			1760–1920			1760–2000		
	Prec	*Recall*	F_1Score	*Prec*	*Recall*	F_1Score	*Prec*	*Recall*	F_1Score
random	0.102	0.511	0.170	0.087	0.483	0.148	0.087	0.503	0.148
Lesk: token overlap	0.234	0.266	0.249	0.245	0.278	0.261	0.248	0.277	0.261
Lesk: sentence embedding	0.269	0.196	0.227	0.266	0.198	0.227	0.280	0.215	0.243
Lesk: w2v	0.323	0.291	0.306	0.288	0.270	0.279	0.286	0.257	0.271
SVM classifier	0.500	0.091	0.155	0.495	0.083	0.143	0.509	0.077	0.133
BERT_base binary centroid	0.254	0.699	0.373	0.238	0.702	0.356	0.236	0.716	0.355
BERT_base sense centroid	0.756	0.464	0.575	0.665	0.471	0.552	0.618	0.493	**0.548**
BERT_base perceptron	0.578	0.425	0.490	0.575	0.448	0.504	0.580	0.456	0.510
BERT_1900 binary centroid	0.234	0.698	0.351	0.221	0.715	0.338	0.222	0.728	0.340
BERT_1900 sense centroid	0.766	0.498	**0.604**	0.702	0.512	**0.592**	0.630	0.497	**0.556**
BERT_1900 perceptron	0.575	0.429	0.492	0.588	0.453	0.511	0.586	0.463	0.517
BERT_1850 binary centroid	0.229	0.678	0.343	0.224	0.713	0.340	0.222	0.722	0.339
BERT_1850 sense centroid	0.789	0.486	**0.602**	0.688	0.500	**0.579**	0.613	0.495	**0.548**
BERT_1850 perceptron	0.587	0.424	0.492	0.568	0.437	0.494	0.570	0.456	0.506

Table 2: Precision, recall and macro F_1scores of the positive class over all senses computed for different time periods. The table highlights the top performing methods for each experiment.

	1850	1920
BERT_base sense centroid	0.575	0.552
BERT_base nearest sense centroid	0.458	0.433
BERT_base weighted sense centroid	0.593	0.556
BERT_1900 sense centroid	0.604	**0.592**
BERT_1900 nearest sense centroid	0.505	0.464
BERT_1900 weighted sense centroid	**0.627**	**0.584**
BERT_1850 sense centroid	0.602	0.579
BERT_1850 nearest sense centroid	0.489	0.441
BERT_1850 weighted sense centroid	**0.609**	0.562

Table 3: Macro Fscores for time (in)sensitive models sense embeddings.

with a corpus spanning more than two centuries.

Experiment 2. Table 3 inspects the performance of the time-sensitive sense embeddings, applying the filtering and weighting to the *sense centroid* methods as explained in Section 7.2. The *weighted* setting is clearly superior compared to *nearest*, and sometimes outperforms the time insensitive approach. However, interestingly, a closer inspection to the results shows that weighting by time can sometimes hinder rather than help, depending on the scenario. Therefore, whereas it clearly seems to help in our experiments in, for example, headwords that largely correspond to abstract senses (*happiness*, *anger*, *art*, *democracy*, *labour*, and *nation*) and in particular for quotations that are further apart from the language model time range (e.g. *BERT_1900* applied to sentences in the 20th century, or *BERT_base* applied to sentences from the 19th and early 20th century), the *weighted* approach seems to be less helpful (and even harmful) with other headwords that have experimented rougher changes in a smaller period of time, such as words from the technological domain (*machine* and *power*) and especially for those quotations that belonged to the the same period as the the language model training or fine-tuning data.

Case studies. The last set of experiments evaluate the merits of our approach in a more focused research scenario, since—as we argued—the task of targeted sense disambiguation is a pragmatic application of word sense disambiguation tailored to the specific research needs of historians and humanities scholars more generally. We report on a series of case studies that group senses in manually curated and meaningful clusters, to simulate how TSD operates as a tool for historical and cultural analysis, for example detecting metaphorical senses of the word *machine* or scrutinizing *power* in the sense of possessing an ability (in contrast to legal interpretations of the term).[24]

As we group multiple senses, we have more examples for each category, meaning that we can evaluate the methods vertically (↓, limited to senses of *one* lemma, i.e. figurative machines versus all other machine senses) and horizontally (→, including the synonyms of the selected senses, i.e. labour in the sense of physical labour and its synonyms such as *work*). Below we report results in both directions, focusing on disambiguating the selected concepts for the long nineteenth century (1760-1920) and only running the most promising models. Before proceeding we should note that, as opposed to previous experiments, the results below have proven more volatile (i.e. dependent on data used in the

[24] All clusters are listed in *run_experiment_curated_cases.py* on the Github repository.

	↓	→
BERT_base sense centroid	0.691	0.536
BERT_base weighted sense centroid	0.582	0.521
BERT_base perceptron	**0.710**	0.493
BERT_1900 sense centroid	**0.700**	0.554
BERT_1900 weighted sense centroid	0.613	**0.566**
BERT_1900 perceptron	0.612	0.526
BERT_1850 sense centroid	0.658	**0.563**
BERT_1850 weighted sense centroid	0.564	0.540
BERT_1850 perceptron	0.621	0.482

Table 4: Macro Fscores for curated case studies.

train and test split) making reproduction trickier.

Not surprisingly, Table 4 indicates that the vertical comparison generally yields slightly higher scores as it is a more constrained task (stays within one lemma). But even after changing the format of the experiments, the results remain fairly consistent with previous findings, the only exception is the high score for the *BERT_base* perceptron, which suddenly achieved a very high precision in the vertical scenario. Nonetheless, the *BERT_1900* model generally has a slight edge over her BERT peers and the sense embedding methods still outperforms other approaches. An additional promising finding for future research is that time-sensitive models do appear as overall very competitive, even obtaining the highest performance for the horizontal experiments. Because these curated experiments are based on a smaller number of examples, results turned out to vary, but future work will look more closely into these distinctive and realistic historical research settings.

9 Conclusion and future work

As language is historically situated, making computational approaches more sensitive to the past should improve performance on semantic tasks relevant to cultural analysis and history.

While the Oxford English Historical Dictionary is undoubtedly a rich resource, the procedure we propose is not confined to English neither does it necessarily require a vast and fine-grained knowledge base as input. Similar dictionaries exist for other languages.[25] Moreover, the method we propose is not necessarily constrained to dictionary data: a particular strength of our approach is that it can learn from a small number of observations. Even with a few carefully collected historical examples, the procedure we propose can be used for exploring senses in a diachronic setting. The OED provided a convenient substitute for the need for annotated examples.

Focusing on targeted sense disambiguation, we demonstrated in this paper that fine-tuning BERT language models on historical texts yields better results, even when including more modern texts in the analysis. Given the complexity of the task and the minimal amount of data to learn from, this suggests that fine-tuning transformers injects historical knowledge in computational models. Historical language models, in combination with the sense centroid method, proved to be a lightweight but efficient tool for exploring the fine-grained semantics of historical texts, which we plan now to adopt to track semantic change at sense level across multiple nineteenth-century textual collections.

More generally, our paper addressed a profound issue: namely how to adapt NLP methods to time. Developing NLP methods more capable to handle the inherent challenges embedded in diachronic data has applications outside of historical and linguistic research and is relevant to the information retrieval and digital libraries communities as well.

[25] E.g. for Italian and for Latin and ancient Greek.

10 Acknowledgements

Work for this paper was produced as part of Living with Machines. This project, funded by the UK Research and Innovation (UKRI) Strategic Priority Fund, is a multidisciplinary collaboration delivered by the Arts and Humanities Research Council (AHRC grant AH/S01179X/1), with The Alan Turing Institute, the British Library and the Universities of Cambridge, East Anglia, Exeter, and Queen Mary University of London. This work was also supported by The Alan Turing Institute under the EPSRC grant EP/N510129/1.

References

Ahmed H. Aliwy and Hawraa A. Taher. 2019. Word sense disambiguation: Survey study. *Journal of Computer Science*, pages 1004–1011.

K. Bhattacharjee, S. ShivaKarthik, S. Mehta, A. Kumar, S. Phatangare, K. Pawar, S. Ukarande, D. Wankhede, and D. Verma. 2020. Survey and gap analysis of word sense disambiguation approaches on unstructured texts. In *2020 International Conference on Electronics and Sustainable Communication Systems (ICESC)*, pages 323–327.

British Library Labs. 2014. Digitised Books. c. 1510 - c. 1900. JSON (OCR derived text). Available via: https://doi.org/10.21250/db14.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Lea Frermann and Mirella Lapata. 2016. A Bayesian model of diachronic meaning change. *Transactions of the Association for Computational Linguistics*, 4:31–45.

Mario Giulianelli, Marco Del Tredici, and Raquel Fernández. 2020. 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. Association for Computational Linguistics.

Hila Gonen, Ganesh Jawahar, Djamé Seddah, and Yoav Goldberg. 2020. Simple, interpretable and stable method for detecting words with usage change across corpora. In *Association for Computational Linguistics*, pages 538–555.

Kristina Gulordava and Marco Baroni. 2011. A distributional similarity approach to the detection of semantic change in the Google Books Ngram corpus. In *Proceedings of GEMS 2011*, pages 67–71. Association for Computational Linguistics.

Kasra Hosseini, Kaspar Beelen, Giovanni Colavizza, and Mariona Coll Ardanuy. 2021. Neural language models for nineteenth-century english. *arXiv preprint arXiv:2105.11321*.

Renfen Hu, Shen Li, and Shichen Liang. 2019. Diachronic sense modeling with deep contextualized word embeddings: An ecological view,. In *ACL 2019*.

C. Kay, J. Roberts, M. Samuels, I. Wotherspoon, and M. Alexander. 2016. Historical thesaurus of english.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Andrey Kutuzov and Mario Giulianelli. 2020. Uio-uva at semeval-2020 task 1: Contextualised embeddings for lexical semantic change detection. In *Proceedings of the 14th International Workshop on Semantic Evaluation, Barcelona, Spain*. Association for Computational Linguistics.

Andrey Kutuzov, Lilja Øvrelid, Terrence Szymanski, and Erik Velldal. 2018. Diachronic word embeddings and semantic shifts: A survey. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1384–1397, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Barbara McGillivray, Simon Hengchen, Viivi Lähteenoja, Marco Palma, and Alessandro Vatri. 2019. A computational approach to lexical polysemy in Ancient Greek. *Digital Scholarship in the Humanities*, 34(4):893–907.

Jean-Baptiste Michel, Yuan Kui Shen, Aviva Presser Aiden, Adrian Veres, Matthew K Gray, Joseph P Pickett, Dale Hoiberg, Dan Clancy, Peter Norvig, Jon Orwant, et al. 2011. Quantitative analysis of culture using millions of digitized books. *science*, 331(6014):176–182.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Sunny Mitra, Ritwik Mitra, Martin Riedl, Chris Biemann, Animesh Mukherjee, and Pawan Goyal. 2014. That's sick dude!: Automatic identification of word sense change across different timescales. In *Proceedings of ACL 2014*, pages 1020–1029.

Roberto Navigli. 2009. Word sense disambiguation: A survey. *ACM Comput. Surv.*, 41(2).

Scott Piao, Fraser Dallachy, Alistair Baron, Jane Demmen, Steve Wattam, Philip Durkin, James McCracken, Paul Rayson, and Marc Alexander. 2017. A time-sensitive historical thesaurus-based semantic tagger for deep semantic annotation. *Computer Speech & Language*, 46:113–135.

Alok Ranjan Pal and Diganta Saha. 2015. Word sense disambiguation: A survey. *International Journal of Control Theory and Computer Modeling*, 5(3):1–16.

Radim Řehůřek and Petr Sojka. 2010. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta. ELRA.

Dominik Schlechtweg, Barbara McGillivray, Simon Hengchen, Haim Dubossarsky, and Nina Tahmasebi. 2020. SemEval-2020 task 1: Unsupervised Lexical Semantic Change Detection. In *Proceedings of the 14th International Workshop on Semantic Evaluation*, Barcelona, Spain. Association for Computational Linguistics.

Dinghan Shen, Guoyin Wang, Wenlin Wang, Martin Renqiang Min, Qinliang Su, Yizhe Zhang, Chunyuan Li, Ricardo Henao, and Lawrence Carin. 2018. Baseline needs more love: On simple word-embedding-based models and associated pooling mechanisms. *arXiv preprint arXiv:1805.09843*.

Sandeep Soni, Lauren F. Klein, and Jacob Eisenstein. 2021. Abolitionist Networks: Modeling Language Change in Nineteenth-Century Activist Newspapers. *Journal of Cultural Analytics*.

Nina Tahmasebi, Lars Borin, and Adam Jatowt. 2018. Survey of computational approaches to lexical semantic change. In *Preprint at ArXiv 2018.*

Nina Tahmasebi and Thomas Risse. 2017. On the uses of word sense change for research in the digital humanities. In *Research and Advanced Technology for Digital Libraries - 21st International Conference on Theory and Practice of Digital Libraries, TPDL 2017, Thessaloniki, Greece, September 18-21, 2017. Proceedings / edited by Jaap Kamps, Giannis Tsakonas, Yannis Manolopoulos, Lazaros Iliadis, Ioannis Karydis*, pages 246–257, Berlin. Springer.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019. HuggingFace's Transformers: State-of-the-art Natural Language Processing. *ArXiv*, abs/1910.03771.