Capturing Evolution in Word Usage: Just Add More Clusters?

Matej Martinc*
matej.martinc@ijs.si
Jozef Stefan Institute
Slovenia

Syrielle Montariol*
syrielle.montariol@limsi.fr
LIMSI - CNRS, Univ. Paris-Sud, Univ. Paris-Saclay, Societé Générale
France

Elaine Zosa*
elaine.zosa@helsinki.fi
University of Helsinki
Finland

Lidia Pivovarova
lidia.pivovarova@helsinki.fi
University of Helsinki
Finland

ABSTRACT

The way the words are used evolves through time, mirroring cultural or technological evolution of society. Semantic change detection is the task of detecting and analysing word evolution in textual data, even in short periods of time. In this paper we focus on a new set of methods relying on contextualised embeddings, a type of semantic modelling that revolutionised the NLP field recently. We leverage the ability of the transformer-based BERT model to generate contextualised embeddings capable of detecting semantic change of words across time. Several approaches are compared in a common setting in order to establish strengths and weaknesses for each of them. We also propose several ideas for improvements, managing to drastically improve the performance of existing approaches.

CCS CONCEPTS

• **Computing methodologies** → **Lexical semantics**; *Cluster analysis*; • **Information systems** → *Language models.*

KEYWORDS

Semantic Change, Contextualised Embeddings, Clustering

ACM Reference Format:
Matej Martinc, Syrielle Montariol, Elaine Zosa, and Lidia Pivovarova. 2020. Capturing Evolution in Word Usage: Just Add More Clusters?. In *Companion Proceedings of the Web Conference 2020 (WWW '20 Companion), April 20–24, 2020, Taipei, Taiwan.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3366424.3382186

1 INTRODUCTION

The large majority of data on the Web is unstructured. Amongst it, textual data is an invaluable asset for data analysts. With the large increase in volume of interaction and overall usage of the Web, more and more content is digitised and made available online, leading to a huge amount of textual data from many time periods becoming accessible. However, textual data are not necessarily homogeneous as they rely on a crucial element that evolves throughout time: language. Indeed, a language can be considered as a dynamic system where word usages evolve over time, mirroring cultural or technological evolution of society [1].

In linguistics, *diachrony* refers to the study of temporal variations in the use and meaning of a word. While analysing textual data from the Web, detecting and understanding these changes can be done for two primary goals. First, it can be used directly for linguistic research or social analysis, by interpreting the reason of the semantic change and linking it to real-world events, and by analysing trends, topics and opinions evolution [9]. Second, it can be used as a support for many tasks in Natural Language Processing (NLP), from text classification to information retrieval conducted on a temporal corpora where semantic change might occur.

To tackle semantic change, models usually rely on word embeddings, which summarise all senses and usages of a word within a certain time period into one vector. Measuring the distance between these vectors across time periods is used to detect and quantify the differences in meaning. But these methods do not take into consideration that most words have multiple senses, since all word usages are aggregated into a single static word embedding. Contextualised embedding models such as BERT [5] are capable of generating a separate vector representation for each specific word usage, making them more suitable for this task.

The goal of this paper is to establish the best way to detect semantic change in a temporal corpus by capitalising on BERT contextualised embeddings. First, several approaches for semantic shift detection from the literature are compared in a common setting in order to establish strengths and weaknesses of each specific method. Second, several improvements are presented, which manage to drastically improve the performance of existing approaches. Our code and models are publicly available[1].

2 RELATED WORK

A large majority of methods for semantic shift detection leverage dense word representations, i.e. embeddings. Word-frequency methods for detecting semantic shift that were popular in earlier studies [13, 16], are now rarely used. The detailed overview of the field could be found in recent surveys [22, 27, 28].

*The authors contributed equally to this research.

WWW '20 Companion, April 20–24, 2020, Taipei, Taiwan

ACM ISBN 978-1-4503-7024-0/20/04.
https://doi.org/10.1145/3366424.3382186

[1] https://github.com/smontariol/AddMoreClusters

2.1 Static Word Embeddings for Semantic Change

The first research that employed word embeddings for semantic shift detection was conducted by [18]. The main idea was to train a separate embedding model for each time period. Since embedding algorithms are inherently stochastic and the resulting embedding sets are invariant under rotation, a procedure that makes these models comparable is needed. To solve this problem, they proposed the incremental model fine-tuning approach, where the weights of the model, trained on a certain time period, are used to initialize weights of a model trained on the next successive time period. Some improvements of the approach were later proposed by [24], who replaced the softmax function for the continuous skipgram model with a more efficient hierarchical softmax, and by [17], who proposed an incremental extension for negative sampling.

An alternative approach was proposed in [19], where embedding models trained on different time periods were aligned in a common vector space after the initial training using a linear transformation for the alignment. The approach was upgraded [31] by using a set of nearest neighbour words as anchors for the alignment.

The third alternative for semantic shift detection with static word embeddings is to treat the same words in different time periods as different tokens in order to get time specific word representations for each time period [6, 26]. Here, only one embedding model needs to be trained and no aligning is needed.

2.2 The Emergence of Contextualised Embeddings

While in static word embedding models each word from the predefined vocabulary is presented as a unique vector, in contextualised embeddings a separate vector is generated for each word mention, i.e. for each context the word appears in. The two most widely used contextual embeddings models are ELMo (Embeddings from LanguageModels [25]) and a more recent BERT (Bidirectional Encoder Representations from Transformers [5]). The approach of using contextual embeddings for semantic shift detection is fairly novel; we are aware of three recent studies that employed it.

In the first study, contextualised embeddings were applied in a controlled way [15]: for a set of polysemic words, a representation for each sense is learned using BERT. Then pretrained BERT is applied to a diachronic corpus, extracting token embeddings, that are matched to the closest sense embedding. Finally, the proportions for each sense are computed at each successive time slice, revealing the evolution of the distribution of senses for each target word. This method requires that the set of senses of each target word is known beforehand.

Another possibility is clustering all contextual embeddings for a target word into clusters representing the word senses or usages in a specific time periods [10]. K-means clustering and BERT contextual embeddings were used in this study. In addition, the incremental training approach proposed by [18] was used for diachronic fine-tuning of the model. Jensen-Shannon divergence (JSD), a measure of similarity between probability distributions, was used to quantify changes between word usages in different time periods. They also tested if domain adaptation of the model would improve the results of their approach by fine-tuning the model on an entire corpus rather than on specific time periods, however this yielded no performance improvements.

In the third, even more recent study, contextual embeddings for a specific word in a specific time period were averaged in order to generate a time specific word representation for each word in each period [23]. BERT embeddings are used in the study and cosine distance is used for measuring the difference between word representations in different time periods.

3 DATA

We rely on a small human-annotated dataset [12] to conduct the evaluation. The dataset consists of 100 words from various frequency ranges, labelled by five annotators according to the level of semantic change between the 1960s and the 1990s. They use a 4-points scale from "0: no change" to "3: significant change", and the inter-rater agreement was 0.51 ($p < 0.01$, average of pair-wise Pearson correlations). The most significantly changed words from the dataset are, for example, *user* and *domain*; words for which the meaning remain intact, are for example *justice* and *chemistry*. This dataset is a valuable resource and has been used to evaluate methods for measuring semantic change in previous research [7, 10]. Following previous work, we use the average of the human annotations as semantic change score. For evaluation, we compute Pearson and Spearman rank correlations between this score and a model output. The notion of the best model is based on Spearman correlations.

To train the models we use the Corpus of Historical American English (COHA) [2]. It contains more than 400 million words of text from the 1810s-2000s. As a historical corpus, it is smaller than the widely used Google books corpus [3] but it has the advantage that data from each decade are balanced by genre—fiction, magazines, newspapers, and non-fiction texts, gathered from various Web sources. We focus our experiments on the most recent data in this corpus, from the 1960s to the 1990s (1960s has around 2.8 million and 1990s 3.3 million words), to match the manually annotated data. The fine-tuning of the model is also done only on this subset.

4 METHODOLOGY

4.1 Context-dependent Embeddings

BERT is a neural model based on the transformer architecture [29]. It relies on a transfer learning approach proposed by [14], where in the first step the network is pretrained as a language model on large corpora in order to learn general contextual word representations. This is usually followed by a task specific fine-tuning step e.g., classification or, in our case, domain adaptation. BERT's novelty is an introduction of a new pretraining learning objective, a *masked language model*, where a percentage of words from the input sequence is masked in advance, and the objective is to predict these masked words from an unmasked context. This allows BERT to leverage both left and right context, meaning that a word w_t in a sequence is not determined just from its left sequence $w_{1:t-1} = [w_1, ..., w_{t-1}]$—as is the case in the traditional language modelling task—but also from its right word sequence $w_{t+1:n} = [w_{t+1}, ..., w_{t+n}]$.

[2] https://www.english-corpora.org/coha/
[3] http://googlebooks.byu.edu/

In our experiments we use the English BERT-base-uncased model with 12 attention layers and a hidden layer of size 768, which was pretrained on the Google Books Corpus [11] (800M words) and Wikipedia (2,500M words). For some of the experiments (see Table 1), we further fine-tune this model (as a *masked language model*) for up to 10 epochs on the COHA subcorpus described in Section 3 for domain adaptation.

Note that our fine-tuning approach deviates from the approaches presented in some of the related work [10] and we do not conduct any diachronic fine-tuning of the model using the incremental training approach similar to [18]. The hypothesis is that this step is not necessary due to contextual nature of embeddings generated by the model, which by definition are dependent on the context that is always time-specific.

Since we are using a pre-trained model we have to apply the BERT tokenization, which is based on byte-pair encodings [30]. In order to acquire contextual embeddings, the corpus documents are first split into sentences; each sentence is limited to 512 tokens and fed into the BERT model. A sequence embedding is generated for each of these sequences by summing last four encoder output layers of BERT[4]. Finally, this sequence embedding of size $sequence\ length \times embeddings\ size$ is cut into pieces, to get a separate contextual embedding for each token in the sequence.

4.2 Target Words Selection

In any practical application of semantic change detection, performing clustering for every word in the corpus would not be feasible in terms of computing time. Thus, we investigate several scalable metrics as a preliminary step to identify a set of words that may have undergone semantic change.

A first set of metrics relies on the computation of a *variation* measure, similarly to [20]. Variation is the cosine distance between each token embedding and a *centroid*, i.e. an average token embedding for a given word. The mean of these cosine distances is the *variation coefficient* of a word. The intuition is that for words that have many different senses and usages, the distance to the centroid would be higher than for words that are monosemous. However, this method does not make distinction between words that gain (loose) sense and polysemous words that stay stable across time.

To measure an evolution of word variation, we compute the variation coefficient inside each time slice t. Then, we take the average difference from one time step to another. This measure aims at detecting words that undergo changes in their level of polysemy. For example, in a corpus divided into T time slices:

$$Variation\ by\ time\ slice = \frac{\sum_{t=t_0}^{T} |Variation_t - Variation_{t-1}|}{T},$$

The second set of metrics relies on *averaging* all token embeddings at each time slice, and using the cosine distance as a measure of semantic drift between time slices. The total drift is the cosine distance between the average of token representations of the first time slice and of the last time slice. It represents the amount of change a word has undergone from the first to the last period, without taking into account the variations in between. The *averaging by time slice* computes the mean of the drifts from each time step to the next one, in order to measure the successive changes of word usage.

To evaluate and compare these measures we use all hundred words from the test set. In practice it is possible to choose a threshold (as a fraction of the size of the full vocabulary) to get a list of target words. Then, the heavier clustering techniques can be applied to this list.

4.3 Embeddings Clustering

The goal of the clustering step is to group the word occurrences by similar vector representation. Then JSD is used to compare cluster distribution across time periods, same as in [10]. The intuition is the following: if, for instance, a word acquired a novel sense in the latter time period, then a cluster corresponding to this sense only consists of word usages from this period but not the earlier ones, which would be reflected by a higher divergence. However, a cluster does not necessarily correspond to a precise sense of the word. Each cluster would rather represent a specific usage or context. Moreover, a word may completely change its context without changing the meaning. Consequently, determining the number of clusters is a tricky part.

For clustering we used k-means with various values for k and affinity propagation [8]. Affinity propagation has been previously used for various linguistic tasks, such as word sense induction [2, 21]. Affinity propagation is based on incremental graph-based algorithm, partially similar to PageRank. Its main strength is that number of clusters is not defined in advance but inferred during training. We also experiment with the approach inspired by [3], where clusters with less than two members are considered weak and merged with the closest strong cluster, i.e. clusters with more than two members.[5] We refer to this method as two-stage clustering.

5 EXPERIMENTS

We focus our analysis on comparing the various clustering approaches and the metrics to detect semantic change. Table 1 shows the Pearson and Spearman correlations between the models' outputs and the human-annotated drifts. We also report Silhouette scores for clustering.

We use a pretrained version of BERT [6] and BERT fine-tuned on the COHA subcorpus for up to 10 epochs. We make use of the Scikit-learn implementation of k-means and affinity propagation [7]. For k-means, we set the number of clusters k and use default parameters for the rest. Similarly, for affinity propagation, we use the default parameters set by the library.

A specificity of BERT is the representation of words with byte-pair encodings [30]. Thus, some words can be divided into several sub-parts; for example, in our list of hundred target words for evaluation, *sulphate* is divided into two byte-pairs *sul* and *##phate*, where *##* denotes the splitting of the word. This is also true for the words *medieval*, *extracellular* and *assay*. We decided to exclude these words from our analysis. Thus, strictly speaking our results

[4] We refer the reader to the original implementation of transformer in [29] for a detailed overview of each component in the architecture.

[5] Note that procedure in [3] is more complex: they first find one or more number of representatives for each datapoint and then clustering is applied over representatives, while in our work clustering is done over the instances themselves.

[6] https://pytorch.org/hub/huggingface_pytorch-transformers/

[7] https://scikit-learn.org/stable/modules/clustering.html

Table 1: Correlations between detected semantic change and manually annotated list of semantic drifts [12] between 1960s and 1990s.

Method	Pearson	Spearman	Silhouette
Related work			
Gulardova & Baroni, 2011 [12]	0.386	-	-
Frermann & Lapata, 2016 [7]	-	0.377	-
Giulianelli, 2019 [10]	0.231	0.293	-
Kutuzov, 2020 [20]	0.233	0.285	-
Pretrained BERT			
Target word selection			
Variation	0.070	0.015	-
Variation by decade	0.239	0.303	-
Averaging by decade	0.295	0.272	-
Averaging	**0.354**	**0.349**	-
Clustering			
k-means, k = 3	0.461	0.444	0.104
k-means, k = 5	0.476	0.443	0.096
k-means, k = 7	0.485	0.434	0.091
k-means, k = 10	0.478	0.443	0.086
2-stage clustering, Aff. propagation	0.530	0.485	-
Affinity propagation	**0.548**	**0.486**	0.039
Fine-tuned BERT for 5 epochs			
Target word selection			
Averaging	0.317	0.341	-
Clustering			
k-means, k=3	0.411	0.392	**0.105**
k-means, k=5	0.539	0.508	0.098
k-means, k=7	0.526	0.491	0.092
k-means, k=10	0.500	0.466	0.088
k-means, k=100	0.315	0.337	0.042
2-stage clustering, Aff. propagation	0.554	0.502	-
Affinity propagation	**0.560**	**0.510**	0.043

are not directly comparable to some of the other approaches in the literature that do not employ BERT.

At the top of Table 1 we overview all previous work on the same test set. To train the models, [13] used GoogleBooks Ngrams, [8] used an extended COHA corpus, and both [11] and [21] used a subcorpus of COHA, identical to the one used in our experiments. In fact, the setting in [11] is quite similar to our work, though our best model performance is much higher than in [11]; we will further discuss this discrepancy in Section 6.

As can be seen in Table 1, among all metrics used for target word selection averaging yields the highest correlation with the human annotations. This intuitively makes sense since averaging measures semantic drift between the first and the last time step and the evaluation dataset was annotated by only considering the first and the last decade. Variation by decade also shows good results; it is a measure of the evolution of the level of variation of a word usage through time.

As can be seen in Table 1 affinity propagation on the fine-tuned BERT model yields the highest Spearman rank correlation. Results obtained using pretrained and fine-tuned models are consistent: in both runs averaging yields lower performance than clustering and affinity propagation is the best clustering method. Two-stage clustering works better than k-means but slightly worse than affinity propagation.

Fine-tuning BERT improves all models except for k-means with 3 clusters and averaging—we do not yet have a clear explanation for that exception.

To conclude, clustering fine-tuned embeddings using affinity propagation yields the best results, with a Pearson correlation with human annotation of 0.56. To evaluate the success of this result, we can use the value of the inter-rater agreement during the annotation process, which was 0.51, computed using the average of pair-wise Pearson correlations [12]. This highlights the difficulty of the task and the performance of the best method.

6 DISCUSSION

6.1 Error Analysis

We manually checked few examples by choosing the words that have less mentions in the corpus to be able to look through all sentences containing the word. One of the tricky cases for our model is the word *neutron*: according to the manual annotation,

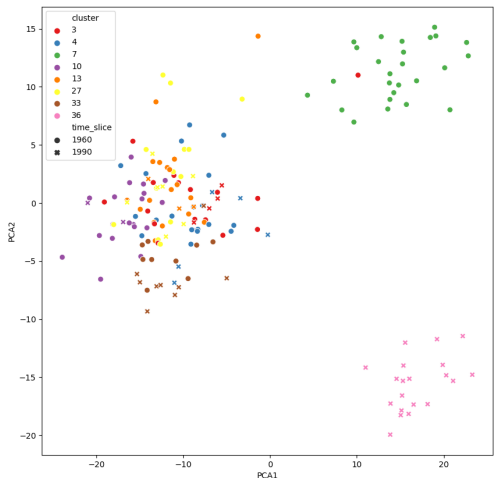

Figure 1: 2D PCA visualization for the biggest clusters obtained for word *neutron*.

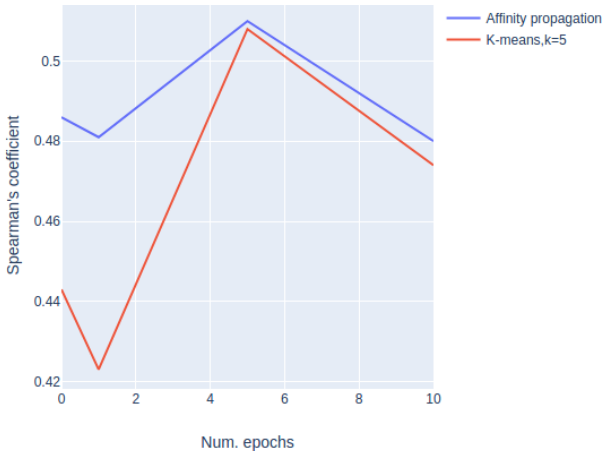

Figure 2: Impact of BERT fine-tuning on the performance of two distinct aggregation methods, affinity propagation and k-means with k=5.

it is ranked 81st and has a stable meaning, while our best model considered it one of the most changed and ranked it at 9.

We visualize the biggest clusters for *neutron* using PCA decomposition of BERT embeddings (Figure 1). There are two clearly distinctive clusters: cluster 36 in the bottom right corner, drawn with pink crosses, which consists only of instances from 1990s, and cluster 7 drawn with green dots in the top right corner, which consists only of instances from 1960s. A manual check reveals that the former cluster consists of sentences which mention *neutron stars*. Though neutron stars have been already discovered in 1960s they were probably less known[8] and are not represented in the corpus. In any case, a difference in a collocation frequency does not mean a semantic shift, since collocations often have a non-compositional meaning. Another similar example is a company called "Vector Security International" that appears only in 1990s time slice, which distorts semantic our calculations for the word *vector*. Our method could be improved by removing stable multiword expressions and named entities from the training set.

The latter distinctive cluster for *neutron*, consisting of word usages from 1960s, contains many sentences that have a certain pathetic style and elevated emotions, such as underlined in the examples below:

throughout the last several decades the dramatic revelation of this new world of matter has been dominated by a most remarkable subatomic particle – the neutron .

the discovery of the neutron by sir james chadwick in 1939. marked a great step forward in understanding the basic nature of matter .

The lack of such examples in 1990s might have a socio-cultural explanation or it could be a mere corpus artefact. In any case, this has nothing to do with semantic shift and demonstrates an ability of BERT to capture other aspects of language, including syntax and pragmatics.

[8]https://en.wikipedia.org/wiki/Neutron_star

6.2 Impact of Fine-tuning

Figure 2 shows the comparison of fine-tuning influence for two best clustering methods (affinity propagation, and k-means with k=5). Interestingly, a light fine-tuning (just for one epoch) decreases the performance of both methods (in terms of Spearman correlation) in comparison to no fine-tuning at all (zero epochs). After that, the length of fine-tuning until up to 5 epochs is linearly correlated with the performance increase.

Fine-tuning the model for five epochs appears optimal. After that, the performance for both methods starts decreasing, most likely because of over-fitting due to the reduced size of the fine-tuning dataset compared to the training data.

The impact of fine-tuning on the k-means clustering is stronger than on the affinity propagation. The difference between model's performance on 5 epochs is negligible. However, this effect holds only with k=5, other values of k do not demonstrate such a difference between original and fine-tuned models, as can be seen in Table 1.

6.3 Clustering

Results presented in Table 1 imply that most of the approaches for semantic change detection proposed in this work manage to outperform previous approaches by a large margin. We believe the differences in the numerical results should be primarily attributed to the differences in the methods, even though we can not draw a direct comparison to some of the approaches due to test set word removal and differences in the train corpora. We can however compare our results directly to the results published by [10] since they are also using BERT trained on the COHA corpus. Even more, their proposed clustering approaches are methodologically very similar to the approaches presented in this work, yet we manage to outperform their approach by a margin of about 35 percentage points when

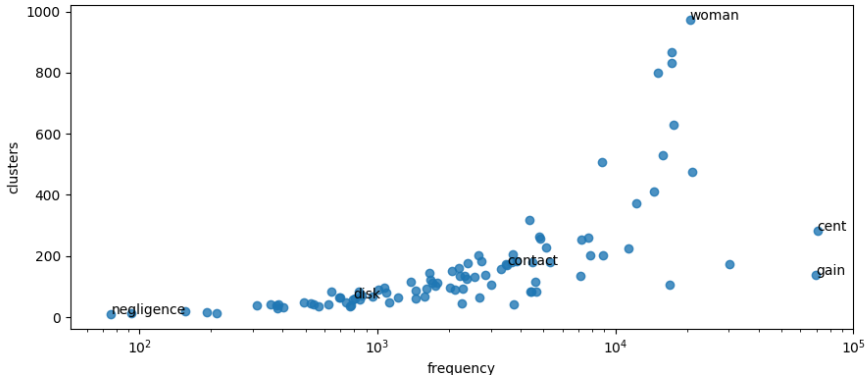

Figure 3: Number of clusters found by affinity propagation and frequency of a word in the 1960s and 1990s in COHA.

affinity propagation is used and by about 33 percentage points when k-means clustering[9], same as in [10], is used.

Unfortunately, [10] does not report a number of clusters that has been used, they only mention that the number of clusters has been optimised using the Silhouette scores. We can only speculate why their results are much lower than ours. The first hypothesis is connected with the usage of the Silhouette score, which might not be optimal for our goals. We compute the Silhouette score[10] for clusterings obtained by our methods. As can be seen in Table 1, the best Spearman correlation coefficient does not correspond to the best Silhouette score. Moreover, the Silhouette scores are quite close to zero.

The second hypothesis is connected with the difference in fine-tuning regimes employed in this research and the one conducted by [10]. We use domain adaptation fine-tuning, proving its efficiency for a certain number of epochs, for both k-means (except for a small number of clusters) and affinity propagation. However, [10] tried both diachronic fine-tuning (using the incremental fine-tuning technique first proposed by [18]) and domain-specific fine-tuning, but concluded that none led to an improvement in the results. As it was already speculated in [10], using both training regimes at the same time might lead to too extensive fine-tuning and therefore over-fitting. Further, a more thorough study on influence of incremental fine-tuning on contextual embeddings models (such as BERT) should perhaps be conducted, since the effects might differ from the ones observed for static embeddings models. Finally, the domain-specific fine-tuning is conducted only for 1 to 3 epochs, which might be too few to improve the results on some corpora.

The difference in performance between k-means and affinity propagation could be partially explained by the different number of clusters in the two approaches. Affinity propagation, which performs the best, outputs a huge amount of clusters—160 on average. The particular number of clusters found by affinity propagation for a word correlates strongly with the frequency of that word in the corpus with correlational coefficient $r = 0.875$, as is illustrated in Figure 3.

Thus, determining the optimal number of clusters for different words is not straightforward. We cannot claim that the clusters found by any of the methods we used can be interpreted as the different senses of a word or that they are even suitable for human interpretation. Most probably, affinity propagation captures subtle differences in word usages rather than global semantic shift. Nevertheless, it works better than k-means with smaller and more intuitive number of clusters, since word sense induction and semantic shift detection are not the same task.

Affinity propagation usually produces a skewed clustering, with a large number of small clusters containing only one or two data points, and can be used for outlier detection. K-means is not suitable for this task since it uses a random initialisation and if an outlier is not initially selected as a potential centroid it may never be found.

To justify this claim we conducted an additional experiment and run k-means clustering on fine-tuned embeddings using k=100 or number of instances minus one for less frequent words. As presented in Table 1, this resulted in Pearson and Spearman rank correlations of 0.315 and 0.337, respectively, which is worse than *any* other strategy we tried for fine-tuned embeddings, including averaging. At the same time, the Silhouette score for this insufficient model is almost equal to the Silhoutte score for the best model. Thus, the Silhouette score fails to discriminate between the best and the worst model.

7 FUTURE WORK

We plan to investigate how the clusters found by the methods in this work can be used to interpret the different usages of a word in a specific time slice. The initial experiments on this subject have already been conducted with the two-stage clustering, which removes the smallest clusters, containing one or two instances. Thus, it allows to focus on a smaller number of the most representative clusters, which might be more suitable for human interpretation even though it does not yield the best result. The initial check demonstrated that most of these clusters are interpretable, though some particular meaning can be spread among several clusters.

[9]Here we are referring to our best k-means configuration with five clusters and using a BERT model fine-tuned for five epochs.
[10]Using standard Scikit-learn implementation, https://scikit-learn.org/stable/modules/clustering.html#silhouette-coefficient

Our analysis hints that clustering BERT token embeddings for a word does not necessarily lead to sense-specific clusters. This conclusion is on par with [4]. Indeed, BERT ability do detect distinct word meanings has limitations. Thus, it would be interesting to extract only the semantic parts of the BERT embeddings to direct our analysis more towards word meaning and rather than word usage in general.

ACKNOWLEDGMENTS

We are grateful to Andrey Kutuzov for valuable discussions during this paper preparation. We also thank Pr. Alexandre Allauzen from ESPCI - Université Paris Dauphine and Pr. Asanobu Kitamoto from National Institute of Informatics (Tokyo) for their advises. This work has been partly supported by the European Union's Horizon 2020 research and innovation programme under grant 770299 (NewsEye) and 825153 (EMBEDDIA).

REFERENCES

[1] Jean Aitchison. 2001. Language Change: Progress Or Decay? In *Cambridge Approaches to Linguistics*. Cambridge University Press, Cambridge.

[2] Domagoj Alagić, Jan Šnajder, and Sebastian Padó. 2018. Leveraging lexical substitutes for unsupervised word sense induction. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

[3] Asaf Amrami and Yoav Goldberg. 2019. Towards better substitution-based word sense induction. *arXiv preprint arXiv:1905.12598* (2019).

[4] Andy Coenen, Emily Reif, Ann Yuan, Been Kim, Adam Pearce, Fernanda B. Viégas, and Martin Wattenberg. 2019. Visualizing and Measuring the Geometry of BERT. In *NeurIPS*.

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. (2019), 4171–4186.

[6] Haim Dubossarsky, Simon Hengchen, Nina Tahmasebi, and Dominik Schlechtweg. 2019. Time-Out: Temporal Referencing for Robust Modeling of Lexical Semantic Change. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 457–470.

[7] Lea Frermann and Mirella Lapata. 2016. A bayesian model of diachronic meaning change. *Transactions of the Association for Computational Linguistics* 4 (2016), 31–45.

[8] Brendan J Frey and Delbert Dueck. 2007. Clustering by passing messages between data points. *Science* 315, 5814 (2007), 972–976.

[9] Nabeel Gillani and Roger Levy. 2019. Simple dynamic word embeddings for mapping perceptions in the public sphere. In *Proceedings of the Third Workshop on Natural Language Processing and Computational Social Science*. 94–99.

[10] Mario Giulianelli. 2019. *Lexical Semantic Change Analysis with Contextualised Word Representations*. University of Amsterdam - Institute for logic, Language and computation.

[11] Yoav Goldberg and Jon Orwant. 2013. A dataset of syntactic-ngrams over time from a very large corpus of english books. In *Second Joint Conference on Lexical and Computational Semantics*. 241–247.

[12] 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 the GEMS 2011 Workshop on GEometrical Models of Natural Language Semantics*. 67–71.

[13] Martin Hilpert and Stefan Th Gries. 2008. Assessing frequency changes in multistage diachronic corpora: Applications for historical corpus linguistics and the study of language acquisition. *Literary and Linguistic Computing* 24, 4 (2008), 385–401.

[14] Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146* (2018).

[15] Renfen Hu, Shen Li, and Shichen Liang. 2019. Diachronic sense modeling with deep contextualized word embeddings: An ecological view. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 3899–3908.

[16] Patrick Juola. 2003. The time course of language change. *Computers and the Humanities* 37, 1 (2003), 77–96.

[17] Nobuhiro Kaji and Hayato Kobayashi. 2017. Incremental Skip-gram Model with Negative Sampling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. 363–371.

[18] Yoon Kim, Yi-I Chiu, Kentaro Hanaki, Darshan Hegde, and Slav Petrov. 2014. Temporal Analysis of Language through Neural Language Models. In *Proceedings of the ACL 2014 Workshop on Language Technologies and Computational Social Science*. 61–65.

[19] Vivek Kulkarni, Rami Al-Rfou, Bryan Perozzi, and Steven Skiena. 2015. Statistically significant detection of linguistic change. In *Proceedings of the 24th International Conference on World Wide Web*. 625–635.

[20] Andrey Kutuzov. 2020. Diachronic contextualized embeddings and semantic shifts. In *press*.

[21] Andrey Kutuzov, Elizaveta Kuzmenko, and Lidia Pivovarova. 2017. Clustering of Russian Adjective-Noun Constructions Using Word Embeddings. In *Proceedings of the 6th Workshop on Balto-Slavic Natural Language Processing*. 3–13.

[22] 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*. 1384–1397.

[23] Matej Martinc, Petra Kralj Novak, and Senja Pollak. 2020. Leveraging Contextual Embeddings for Detecting Diachronic Semantic Shift. In *LREC*.

[24] Hao Peng, Jianxin Li, Yangqiu Song, and Yaopeng Liu. 2017. Incrementally learning the hierarchical softmax function for neural language models. In *Thirty-First AAAI Conference on Artificial Intelligence*.

[25] Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep Contextualized Word Representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. 2227–2237.

[26] Alex Rosenfeld and Katrin Erk. 2018. Deep neural models of semantic shift. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. 474–484.

[27] Nina Tahmasebi, Lars Borin, and Adam Jatowt. 2018. Survey of Computational Approaches to Diachronic Conceptual Change. *arXiv preprint arXiv:1811.06278* (2018).

[28] Xuri Tang. 2018. A state-of-the-art of semantic change computation. *Natural Language Engineering* 24, 5 (2018), 649–676.

[29] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*. 5998–6008.

[30] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144* (2016).

[31] Yating Zhang, Adam Jatowt, Sourav S Bhowmick, and Katsumi Tanaka. 2016. The past is not a foreign country: Detecting semantically similar terms across time. *IEEE Transactions on Knowledge and Data Engineering* 28, 10 (2016), 2793–2807.