Lexicographical Semantic Change Detection with BERT

Salih Furkan Akkurt
Boğaziçi University
Department of Computer Engineering
34342 Bebek, Istanbul, Turkey
furkan.akkurt@boun.edu.tr

Abstract

I publicly release the code and references in a repository (Akkurt, 2023).

1 Introduction

Dictionaries are great sources of information related to languages. They contain a lot of information regarding meanings (senses) of words. As a language evolves, senses of its words change and dictionaries need to be updated accordingly. Dictionary writers constantly work on manually detecting semantic changes in current usage. This is a very time-consuming and expensive process.

In this work, I propose a method to automatically detect semantic changes in the English language by leveraging the contextualizing power of deep learning-based language models. I use BERT (Devlin et al., 2019) to contextualize words in two corpora and cluster their contextualized representations to detect semantic changes. The current proposed method in this paper does not detect if a sense of a word morphed into another sense but detects if a word has gained a new sense or lost one. This method can be extended into detecting morphing of senses also.

2 System Description

2.1 Data

I use 3 types of data in my work. The first type is a frequency list of words, the second a dictionary and the third corpora.

Firstly, I use the frequency list of words (Davies, 2023b) from the Corpus of Contemporary American English (COCA) (Davies, 2008). This list serves as my vocabulary which I use to detect semantic changes in. I restrict my vocabulary to *nouns* only. There are 2635 nouns in the list, the most frequent three of which are *man*, *case* and *money*.

Secondly, I use Merriam-Webster (MW) (Merriam-Webster, 2023a) as my dictionary. Merriam-Webster is a well-known dictionary that is updated quite frequently. I have used its API (Merriam-Webster, 2023b) to get the sense counts of the words in my vocabulary. These sense counts include all entries of the words, meaning with any parts of speech, not just nouns. They serve two purposes: (1) to determine the semantic distance of the embeddings that is necessary for a sense distinction in dictionaries and (2) to, in the end, compare the number of senses of the words in the dictionary with the cluster counts of the word embeddings in the corpora.

Lastly, I use free samples (Davies, 2023a) of the COCA and the NOW corpus (News on the Web) (Davies, 2016) as my two corpora that serve as the current usage of the language. The COCA's sample has 8.9 million words of linear tokenized text and the one of the NOW's has 1.7 million words. Content of the COCA ranges from academic usage to fiction and spoken language. The NOW corpus is a collection of web-based news sources. Since the task is detecting current semantic changes, the corpora should be recent.

2.2 Method

Firstly, the two corpora is merged into one. When a given sentence is passed through BERT, each token of the sentence has its own contextual representation, as demonstrated by (Pasini et al., 2020). I use BERT (Devlin et al., 2019) to contextualize words in the merged corpus and gather token representations of each word. The implementation of BERT (base-uncased) on HuggingFace's Transformers (Wolf et al., 2020) is used. After this step, each word has its own sets of representations. (Zhou and Li, 2020) used "the sum of the last 4 layers [of BERT] to encode both word meaning and context information". Instead of summation, each representation is the concatenation of the last 4 layers of BERT's output in this work. The representations are stored after a rounding operation to reduce the size of the data as the concatenation increases the size of the data by 4 times as opposed to summation. Experiments showed that the rounding operation does not affect the results significantly.

After the contextualization step, the agglomerative clustering algorithm (Florek et al., 1951) is used to cluster the representations of each word. The implementation of the algorithm on scikit-learn (Pedregosa et al., 2011) requires the number of clusters or the distance threshold to be specified. I use the number of senses of the words in the dictionary as the number of clusters input to the algorithm. The cosine distance metric is used to calculate the distances between these clusters. The minimum distance between the clusters is then used as the distance threshold in the second clustering step. This distance represents the semantic distance that compelled the dictionary writers to distinguish the senses of the word.

The second clustering step is the same with the first one except the distance threshold is used instead of the number of clusters. This step is used to create new clusters without specifically knowing the number of clusters beforehand but the distance threshold. In the agglomerative clustering algorithm, the distance threshold is the maximum distance between two clusters to be merged. In the end, the number of clusters is the number of senses of the word represented in the corpora.

If the number of clusters of the second clustering is greater than the number of senses of the word in the dictionary, it means that the word has gained a new sense. If the number of clusters is less than the number of senses of the word, it means that the word has lost a sense. However, if the number of clusters is equal to the number of senses of the word, it does not mean that the word has not changed semantically, as it may have gained a sense and lost another at the same time.

3 Experiments and Results

3.1 Cosine distance for semantic change detection

First, I experimented on whether the cosine distance between the BERT representations of the words in the corpora can be used to detect semantic changes in the words. I constructed two paragraphs with the word *bank*, differing only in the middle sentences, one representing the financial sense of the word and the other the river bank sense.

- First paragraph: I wake up early in the morning and start my day by having a cup of coffee and reading the news. After that, I take a shower and get dressed for work. I work from 9 to 5, and my job keeps me busy throughout the day. During my lunch break, I usually go for a walk in the park or grab a quick bite to eat at a nearby café. Today, I need to withdraw some cash, so I decide to go to the *bank* during my lunch break. I walk to the *bank*, which is just a few blocks away from my office, and withdraw the money that I need. After that, I grab a sandwich and a soda and head back to the office to finish my work for the day. When I get home in the evening, I like to unwind by watching TV or reading a book. After dinner, I usually spend some time chatting with my family or friends before going to bed.

- Second paragraph (only the differing part): Today, since the weather is so hot, I decide to go to the river *bank* to swim during my lunch break.

The cosine distances between the representations of the word *bank* in the two paragraphs are calculated. Distances between usages:

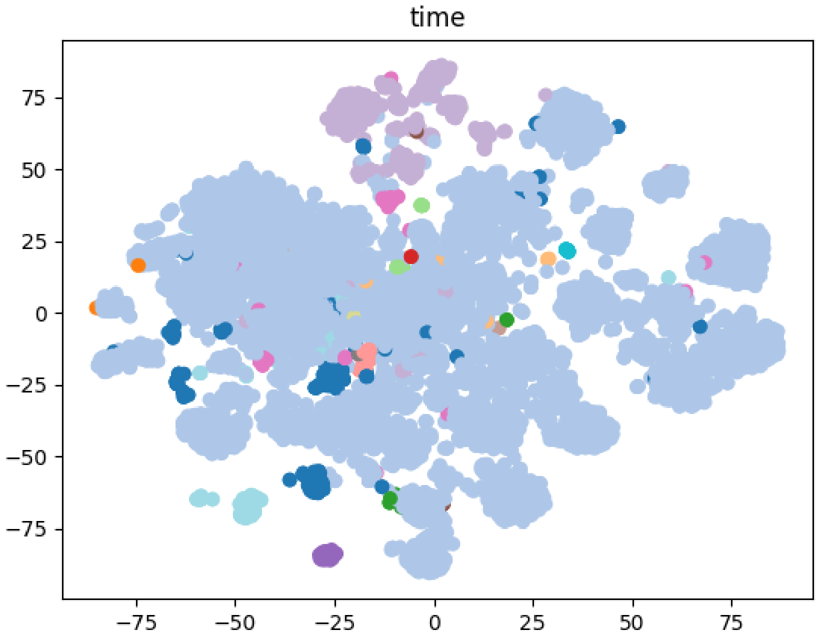

Figure 1: The T-SNE plot of the contextualized representations of the word *time* after the first clustering (number of clusters known) step.

- Two usages in the first paragraph: 0.2214
- First usage in the first paragraph and the only usage in the second paragraph: 0.3990
- Second usage in the first paragraph and the only usage in the second paragraph: 0.4330

More experiments between several sentences with the same and different usages of words were conducted but the results are not consistent. The cosine distances between the token representations are not reliable enough to detect semantic changes in the words. Even though this is the case, I continued on to the next step of the method.

3.2 Semantic change detection with BERT and clustering

I have implemented the method described in the previous section completely and run it all the way through. Minimum cosine distance between any 2 clusters for all the embeddings of the entire vocabulary turns out to be 0.0645. Using this minimum distance as the distance threshold, I have run the second clustering step. This step yielded 11,081 clusters for the word *time*. The number of senses of the word *time* in the dictionary is 23. This means that the minimum distance that compelled the dictionary writers to create different senses in the dictionary is not representative of the entire semantic space. If we use this small distance to cluster the embeddings, we end up with thousands of clusters for a single word, which is not useful for our purposes.

I have generated 2 T-SNE plots of the contextualized representations of the word *time* after the first and second clustering steps, labels coming from the agglomerative clustering algorithm. The first plot is shown in Figure 1 and the second one in Figure 2.

4 Discussion and Conclusion

This work is a preliminary work on the topic of semantic change detection in lexicography. The results are not very promising but the method is quite simple and can be improved upon. The method can be improved by using a better clustering algorithm and a better distance metric.

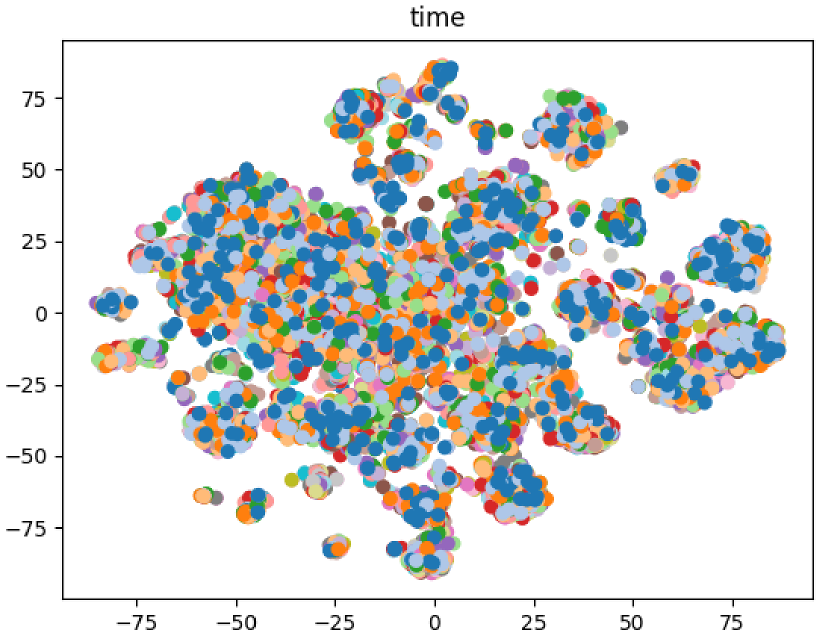

Figure 2: The T-SNE plot of the contextualized representations of the word *time* after the second clustering (distance threshold known) step.

For future work, with higher computational power, the whole COCA and NOW corpora instead of their samples can be used to represent a broader range of the current usage of the language. Also, the vocabulary can be expanded to include other parts of speech such as verbs and adjectives.

To conclude, in this paper, I proposed a method to automatically detect semantic changes in words. This method uses BERT to contextualize words in raw text and then clusters their contextualized representations to compare against the sense counts of the words in the dictionary. As languages continuously change and dictionaries need updates quite frequently, this work can be used to help dictionary writers detect changes in language more efficiently.

References

Furkan Akkurt. 2023. furkanakkurt1335/58t-app: Application project for CMPE58T. GitHub repository.

Mark Davies. 2008. The corpus of contemporary american english (COCA). Online. [Online at `https://www.english-corpora.org/coca/` ; accessed 30 May 2023].

Mark Davies. 2016. The NOW corpus (news on the web). Online. [Online at `https://www.english-corpora.org/now/` ; accessed 30 May 2023].

Mark Davies. 2023a. Full-text data from english-corpora.org: billions of words of downloadable data. Online. [Online at `https://www.corpusdata.org/formats.asp` ; accessed 30 May 2023].

Mark Davies. 2023b. Word frequency: based on one billion word COCA corpus. Online. [Online at `https://www.wordfrequency.info/samples.asp` ; accessed 30 May 2023].

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, June. Association for Computational Linguistics.

K. Florek, Józef Łukaszewicz, J. Perkal, Hugo Steinhaus, and S. Zubrzycki. 1951. Sur la liaison et la division des points d'un ensemble fini. *Colloquium Mathematicum*, 2:282–285.

Incorporated Merriam-Webster. 2023a. Merriam-webster. Online. [Online at `https://www.merriam-webster.com`; accessed 30 May 2023].

Incorporated Merriam-Webster. 2023b. Merriam-webster dictionary API. Online. [Online at `https://dictionaryapi.com`; accessed 30 May 2023].

Tommaso Pasini, Federico Scozzafava, and Bianca Scarlini. 2020. CluBERT: A cluster-based approach for learning sense distributions in multiple languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4008–4018, Online, July. Association for Computational Linguistics.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, October. Association for Computational Linguistics.

Jinan Zhou and Jiaxin Li. 2020. TemporalTeller at SemEval-2020 task 1: Unsupervised lexical semantic change detection with temporal referencing. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 222–231, Barcelona (online), December. International Committee for Computational Linguistics.