Temporal Word Meaning Disambiguation using TimeLMs

Mihir Godbole[*] and **Parth Dandavate**[*] and **Aditya Kane**[*]
Pune Institute of Computer Technology, Pune
{mihirgod11, dandavateparth, adityakane1}@gmail.com

Abstract

Meaning of words constantly change given the events in modern civilization. Large Language Models use word embeddings, which are often static and thus cannot cope with this semantic change. Thus,it is important to resolve ambiguity in word meanings. This paper is an effort in this direction, where we explore methods for word sense disambiguation for the EvoNLP shared task. We conduct rigorous ablations for two solutions to this problem. We see that an approach using time-aware language models helps this task. Furthermore, we explore possible future directions to this problem.

1 Introduction

A change in the meaning of a word in varying semantic contents is a challenge for various NLP tasks such as text and sentence classification, question answering and sentence prediction. Recent developments in large language models (LLMs) like ELMo (Peters et al., 2018), BERT (Devlin et al., 2019) and GPT (Brown et al., 2020) have revolutionised the field of NLP with context dependent word embeddings. These models have been trained on a large corpus of unlabelled text. While these models take in consideration the semantics of the text, it is limited to the corpus it was trained on. This introduces a new challenge of the shift in the meaning of a word across the temporal axis.

Word Sense Disambiguation (Huang et al., 2019) is the process of identifying the meaning of a word from multiple possible meanings in varying contexts. This task can be further extended as a polysemy resolution task to classify the meaning of words in different contexts. Our system performs a similar task while classifying two texts with a common word with the same or different meaning. Specifically, the premise of our system is to classify tweets from two different time periods with a common word. The variation in the meaning of a word is caused by two factors, the context of the word in the form of a tweet or a change in the usage and hence in the meaning of a word because of the shift along the time axis. Historically it has been observed that the meanings of some words have been altered over time. For example, the word "fathom" originally meant "to encircle with one's arms” and now is defined as “to understand after much thought”. The ever expanding nature of the internet and social media have led to rapid evolution of words, with the meanings of words changing and new words getting csoined. This means that the corpus of data used for training a LLM will keep changing over time. Hence, the pretrained models for existing LLMs like BERT, RoBERTa cannot be used to compare word embeddings for a word from two different time periods. This shared task (Loureiro et al., 2022b) focusses precisely on this problem statement.

To address this problem, we propose a system comprising of TimeLMs (Loureiro et al., 2022a) to incorporate the time aspect of the data. TimeLMs are language models that are trained using data up to a certain time instance. In this case they are trained on tweets gathered by the end of a year. Therefore there exists a unique TimeLM model for each year which takes into account the time aspect of data. The dataset used for testing our system consists of tweets from the years 2019, 2020 and 2021. Tweets from two different time periods containing a common word are paired in this dataset and labelled to indicate similarity or dissimilarity in the meanings of that word in the two tweets. The TimeLMs used in our system are Roberta models trained on tweets upto the specific year. This enables our system to get an accurate representations of the words based on their use upto that time period. The embeddings are then compared based on a similarity metric to classify the tweets using a preset threshold value for similarity.

[*] Equal Contribution

Proceedings of the The First Workshop on Ever Evolving NLP (EvoNLP), pages 55 - 60
December 7, 2022 ©2022 Association for Computational Linguistics

This paper is organised as follows. We analyse existing research and methods in Section (2). We give a overview of the dataset used for our system in Section (3). We provide a overview of our system implementation in Section (4). We also compare the results of our experiments in developing this system in Section (5). We discuss the possible improvements and scope of this system in Section (6).

2 Related work

In Natural Language Processing, the meaning of words is denoted by a vector, commonly known as word embedding. Works like GloVe (Pennington et al., 2014) and Word2Vec (Mikolov et al., 2013) were one of the first ones to represent a word using vectors. However, the embeddings thus generated were context-agnostic, meaning their meaning was fixed and were not dependent on the context.

With the dawn of modern text encoders (Vaswani et al., 2017; Devlin et al., 2019), context dependent embeddings can be easily calculated. Works like Pilehvar and Camacho-Collados; Raganato et al. aim to have manually annotated datasets containing pairs of sentences having same or different meaning, and labelling them as such. To solve this task, several methods have been developed. Works like Levine et al. (2020); Peters et al. (2019) try to impart context based knowledge into the embeddings by using WordNet (Miller, 1995) attributes. The models are trained in a self-supervised fashion with entity linking. Another approach is to use word-level embeddings. Loureiro and Jorge (2019) use this approach, combining it with a k-NN (k Nearest Neighbours) method to disambiguate the word embeddings. Note that transformers can also be used for this purpose, since the output features from the transformers can be interpreted as word embeddings. Loureiro et al. (2022c) studies model layers to understand the effect of attention-based architectures in word sense disambiguation task. Elmo (Peters et al., 2018) is one of many available architectures in this direction. Lastly, work has been done to incorporate the semantic space knowledge into the embeddings (Colla et al., 2020), also known as sense-based disambiguation.

Given this, little work has been done on word meaning disambiguation in a temporal setting. This means that the information about the time of text utterance is also provided along with the sentence itself. This paper tries to provide a solution to this problem - word meaning disambiguation when temporal information is available.

3 Dataset description

The dataset consists of 1428 training samples and 396 validation samples. The final scores were calculated on a set of 10,000 unseen test samples. In every training sample, we were provided with two sentences and the word whose semantic meaning was to be compared. Some metadata like tokens and start and end of word was also included in every sample.In the training dataset, out of the 1428 samples, 650 examples had the words in two sentences having same meaning, whereas 778 samples had the words in two sentences having different meaning. Note that since this dataset is relatively balanced, and hence does not need any additional preprocessing to balance the data distribution. However, it is important to note that the target words in the training and testing dataset constitute two different sets, and hence the problem should be solved in a way that is target word agnostic.

An illustration of the data is shown in Figure 1. The left part shows an example where the meaning of the target word "virus" is different in both tweets. Specifically, in the top left tweet it indicates to the disease-causing organism whereas the bottom left tweet indicates to a thing that the person likes. In the right part, both instances of the target word mean the same, denoting disease-causing organism.

The dataset also provides the month and year when the tweet was written. This provides us the temporal information, which can be useful for the semantic evaluation of the words in the given context. Our approach aims at using this semantic information in a way that a language model relevant to the tweet is used to get the semantic features of the tweet.

4 Methodology

4.1 TimeLMs aided word sense disambiguation

As mentioned in Section 3, the date of posting of the tweet is provided as a data attribute. In this approach, we used this information to choose the transformer model to extract target features. We use TimeLMs (Loureiro et al., 2022a) for this purpose. We observe this performs better compared to using a single model. Our method is illustrated in Figure 2.

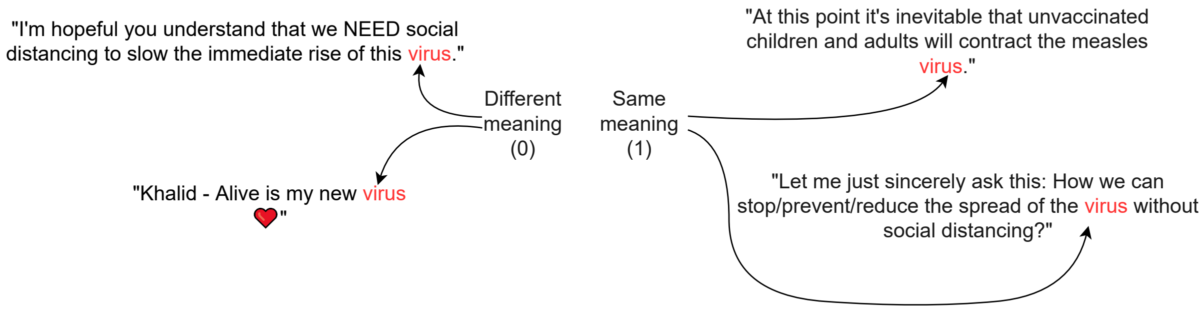
"I'm hopeful you understand that we NEED social distancing to slow the immediate rise of this virus."
"Khalid - Alive is my new virus ❤"
Different meaning (0)
Same meaning (1)
"At this point it's inevitable that unvaccinated children and adults will contract the measles virus."
"Let me just sincerely ask this: How we can stop/prevent/reduce the spread of the virus without social distancing?"

Figure 1: Examples from the datset.

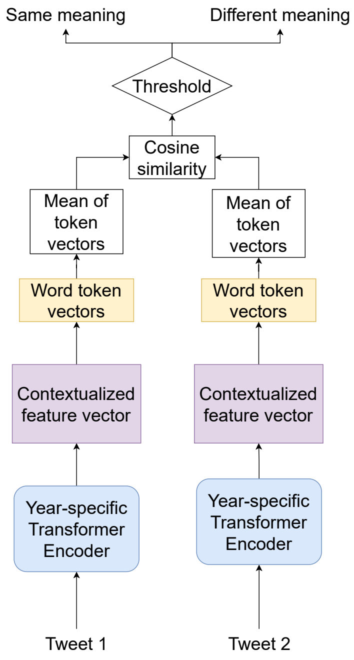

Figure 2: TimeLMs aided word sense disambiguation

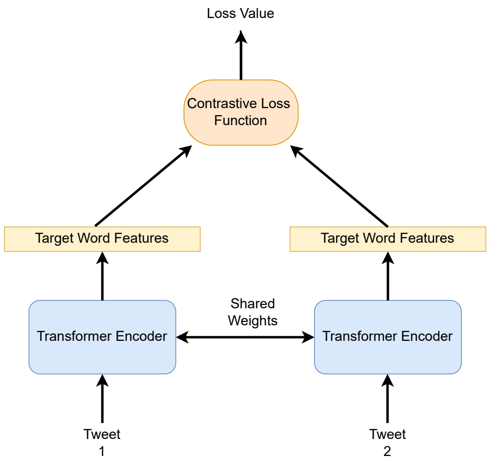

Figure 3: Contrastive feature based classification

Specifically, we observe that the tweets in the input data are posted in the years 2019 and 2020 only. Thus, we use the variants of TimeLMs trained on Twitter data collected until December 2019 and 2020 for respectively dated tweets. In this way, we can encapsulate the difference in semantic representations of sentences across time.

After extracting the contextualized sentence features from the respective models, we extract the target word features. We hereby get two word feature vectors, one corresponding to each tweet. Note that since one word may be split into multiple tokens, we use the mean of these token-wise features for out computation. Note that the feature vectors for a tweet is the mean of the last four layers of the language models concatenated to the pooled ($[CLS]$ token) output. These two feature vectors are then compared with each other using cosine similarity. If this cosine similarity is high, the meaning of the target word in two sentences is the same, alternatively if the cosine similarity is low then the meaning of the target word in the two sentences is different.

Since this approach does not actually train the parameters of the model, we use the training dataset to calculate the thresholds. Specifically, we iterate over potential thresholds between 0 and 1 with a step of 0.001. We then rank the thresholds based on their F1 scores. The best performing threshold is then used for generating the final predictions. The threshold for our best performing model (TimeLMs) was 0.917. We use five models for our ablations: ELECTRA (small) (Clark et al., 2020), ALBERT (base) (Lan et al., 2019), BERT (base, uncased) (Devlin et al., 2019), RoBERTa (base) (Zhuang et al., 2021), TimeLMs (Loureiro et al., 2022a).

4.2 Contrastive feature based classification

The task essentially being identifying whether the usage of word is similar or not we thought of training the language models in a Siamese setting. Siamese networks involves two similar encoder networks with the same weights and a classification system, which determines the similarity based on the distance between encoded features and a threshold. As mentioned in the previous sub section we are extracting the target word features using transformer models which will be the encoders. If the meaning of the word in both the sentences is same then the target word features given by the transformer model should be similar.

We trained the model using a simple contrastive loss involving euclidean distance between the target word features. We used the same models as mentions in the previous sections, except for the TimeLMs. For determining the threshold for the classification process we iterated through a range of 0 to 4, with a step of 0.01, while testing on the validation data. The threshold was determined for the euclidean distance between the word embeddings obtained from the model. The threshold for our best performing model (TimeLMs) was 1.148.

4.3 Implementation details

We use the HuggingFace library (Wolf et al., 2020) for our experiments. For the cosine similarity experiments, we find a threshold of 0.917 for our best performing solution. We use a batch size of 64. Here the threshold was selected for the cosine distance between the two word embeddings.

For the contrastive method experiments, we find a threshold of 1.148 for our best performing solution (RoBERTa). We used a batch size of 8. Here the threshold was selected for the euclidean distance between the two word embeddings.

In both cases, the inference distance value (cosine or euclidean) below the threshold indicated similar meaning for the two words, and the the inference distance value above the threshold indicated different meaning for the two words.

5 Results

We hereby present the results of both of our methods. We report several interesting observations based on the results.

Our results based on our cosine similarity are shown in Table 1 and our results based on the contrastive method are shown in Table 2.

Model	Val F1-score	Val Accuracy	Test F1-score
Electra	61.00	54.78	38.77
RoBERTa	60.00	56.51	38.96
BERT	60.80	56.77	38.77
Albert	60.73	56.77	39.00
TimeLMs	**61**	**61.71**	**57.94**

Table 1: Results of Similarity Method

Model	Val F1-score	Val Accuracy	Test F1-score
Electra	**66.67**	**75**	46.15
RoBERTa	60.8	44.01	**48.97**
BERT	65.44	48.98	44.34
Albert	66.6	66.6	43.75

Table 2: Results of Contrastive Method

1. **TimeLMs based method performs the best:** We observe that the TimeLMs based method performs the best. We speculate this is because of the time-aware nature of the models. Some words, for example "lockdown" have significantly different meaning before and after the pandemix. Thus, models pretrained on the specific data results in better performance.

2. **BERT and AlBERT have similar performance:** We see that BERT and Albert have very similar Accuracy and Macro-F1. We hypothesize that this is because of the similarity in their pretraining objectives. Albert is a model aimed to mimic the capabilities of BERT, but with lower number of parameters. Thus, it makes sense that these models have very similar validation metrics.

3. **Electra has a better language representation:** As seen on state of art benchmarks like GLUE and SQuAD Electra is outperforming RoBERTa, ALBERT. Electra has achieved better F1-score and accuracy compared to both.

6 Conclusion

In this paper, we explore two solutions to the word sense disambiguation problem within the scope of EvoNLP shared task. We report a maximum testing F1-score of 57.94% with TimeLMs. We foresee several research directions for this work. One line of work can be explore robustness of the contrastive models. The threshold search technique for this method can be explored in greater detail.

References

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: pre-training text encoders as discriminators rather than generators. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Davide Colla, Enrico Mensa, and Daniele P. Radicioni. 2020. LessLex: Linking multilingual embeddings to SenSe representations of LEXical items. *Computational Linguistics*, 46(2):289–333.

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.

Luyao Huang, Chi Sun, Xipeng Qiu, and Xuanjing Huang. 2019. GlossBERT: BERT for word sense disambiguation with gloss knowledge. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3509–3514, Hong Kong, China. Association for Computational Linguistics.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations.

Yoav Levine, Barak Lenz, Or Dagan, Ori Ram, Dan Padnos, Or Sharir, Shai Shalev-Shwartz, Amnon Shashua, and Yoav Shoham. 2020. SenseBERT: Driving some sense into BERT. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4656–4667, Online. Association for Computational Linguistics.

Daniel Loureiro, Francesco Barbieri, Leonardo Neves, Luis Espinosa Anke, and Jose Camacho-collados. 2022a. TimeLMs: Diachronic language models from Twitter. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 251–260, Dublin, Ireland. Association for Computational Linguistics.

Daniel Loureiro, Aminette D'Souza, Areej Nasser Muhajab, Isabella A. White, Gabriel Wong, Luis Espinosa-Anke, Leonardo Neves, Francesco Barbieri, and Jose Camacho-Collados. 2022b. TempoWiC: An evaluation benchmark for detecting meaning shift in social media. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 3353–3359, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Daniel Loureiro and Alípio Jorge. 2019. Language modelling makes sense: Propagating representations through WordNet for full-coverage word sense disambiguation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5682–5691, Florence, Italy. Association for Computational Linguistics.

Daniel Loureiro, Al'ipio M'ario Jorge, and José Camacho-Collados. 2022c. Lmms reloaded: Transformer-based sense embeddings for disambiguation and beyond. *Artif. Intell.*, 305:103661.

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

George A. Miller. 1995. Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Matthew E. 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)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Matthew E. Peters, Mark Neumann, Robert Logan, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A. Smith. 2019. Knowledge enhanced contextual word representations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 43–54, Hong Kong, China. Association for Computational Linguistics.

Mohammad Taher Pilehvar and José Camacho-Collados. 2018. Wic: 10, 000 example pairs for evaluating context-sensitive representations. *CoRR*, abs/1808.09121.

Alessandro Raganato, Tommaso Pasini, José Camacho-Collados, and Mohammad Taher Pilehvar. 2020. Xl-wic: A multilingual benchmark for evaluating semantic contextualization. *CoRR*, abs/2010.06478.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

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. Association for Computational Linguistics.

Liu Zhuang, Lin Wayne, Shi Ya, and Zhao Jun. 2021. A robustly optimized BERT pre-training approach with post-training. In *Proceedings of the 20th Chinese National Conference on Computational Linguistics*, pages 1218–1227, Huhhot, China. Chinese Information Processing Society of China.