

Analysis of the Semantic Vector Space Induced by a Neural Language Model and a Corpus

Xinying Chen¹, Jan Hůla¹ and Antonín Dvořák¹

¹*Institute for Research and Applications of Fuzzy Modeling, University of Ostrava, CE IT4Innovations, 30. dubna 22, 701 03 Ostrava, Czech Republic*

Abstract

Although contextual word representations produced by transformer-based language models (e.g., BERT) have proven to be very successful in different kinds of NLP tasks, there is still little knowledge about how these contextual embeddings are connected to word meanings or semantic features. In this article, we provide a quantitative analysis of the semantic vector space induced by the XLM-RoBERTa model and the Wikicorpus. We study the geometric properties of vector embeddings of selected words. We use HDBSCAN clustering algorithm and propose a score called Cluster Dispersion Score which reflects how disperse is the collection of clusters. Our analysis shows that the number of meanings of a word is not directly correlated with the dispersion of embeddings of this word in the semantic vector space induced by the language model and a corpus. Some observations about the division of clusters of embeddings for several selected words are provided.

Keywords

semantic vector space, neural language models, vector embeddings, clustering analysis, polysemy

1. Introduction

Contextual word representations (embeddings) produced by transformer-based language models, such as BERT, have proven to be valuable and very successful in different kinds of NLP tasks, including machine translation, text generation, word sense disambiguation, etc. However, there is still little knowledge about how these contextual embeddings are connected to word meanings or semantic features.

We believe that if we better understand the relation of these embeddings to semantics of corresponding words, we will be able to figure out the way in which transformer-based models learn and represent natural language. It can also help to design more robust methods for word sense disambiguation, analysis of semantic change, and related tasks.

In this article, we provide a quantitative analysis of the semantic vector space induced by a popular language model called XLM-RoBERTa [1] and a text corpus called Wikicorpus [2]. Concretely, we study the geometric properties of vector embeddings of selected polysemous (e.g., “developer”) and monosemous (e.g., “sheet”) words.¹ For a given word, we collect all sentences containing this word, process these sentences by the language model, and collect word-specific embeddings. We then used the UMAP algorithm to reduce the dimensionality of the em-

beddings and apply the HDBSCAN clustering algorithm to cluster these embeddings.

To study the geometric properties of this collection of clusters of word-specific embeddings, we propose a measure called Cluster Dispersion Score. We provide figures and descriptions of the results for several selected words. We also quantify the correlation between the score and the number of meanings of a given word. Our analysis shows that the number of meanings of a word is not directly correlated with the dispersion of the embeddings of this word in the semantic vector space induced by the language model and a corpus.

The paper is structured as follows. Section 2 discusses related work on the usage and properties of embeddings obtained by transformer models. In Section 3, we describe the methods we use, including the selection of words we investigate, the computation of embeddings, clustering, the computation of the cluster dispersion score, and cluster summarization. The description of our experiments and results can be found in Section 4. It also contains a more detailed description of the results for several selected target words. Then, a discussion of the interpretation of the results is provided in Section 5. Finally, Section 6 contains conclusions and directions for further research.

2. Related Work

Although neural network language models are well recognized for their ability to capture contextual semantics, in-depth discussions about the relationships between word vector representations and word meanings are not so common. The majority of works are concentrated on


ITAT'22: Information technologies – Applications and Theory, September 23–27, 2022, Zuberec, Slovakia

✉ cici13306@gmail.com (X. Chen); jan.hula@osu.cz (J. Hůla);

antonin.dvorak@osu.cz (A. Dvořák)

© 2022 Copyright for this paper by its authors. Use permitted under Creative Commons License

Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

¹For details on how we differentiate between monosemous and polysemous words see Section 3.1.

improving the performance of language models for Word Sense Disambiguation (WSD) tasks, and only a few are investigating how language models encode and recover word senses.

As a semantic disambiguation task [3], WSD has progressed greatly since the appearance of neural network language models [4]. This is especially true for transformer-based models [5]. For instance, BERT and its derivatives (BERT family models) have proven to be very successful for WSD and word embeddings produced by these models can deliver rather satisfying results even with a simple non-parametric approach (e.g., nearest neighbors) and a small training set [6, 7]. However, with the priority of improving WSD performance, such studies offer little insight into word vector organizations.

A few works have attempted to discuss more in-depth how transformer-based language models encode semantic knowledge, such as semantic information provided by WordNet (a predefined word sense inventory). Loureiro et al. [7] provided quantitative and qualitative analysis of different classes of words (with different numbers of meanings) in the BERT model and found that BERT can capture high-level or coarse-grained sense distinctions, but it does not capture fine-grained sense distinctions. In reality, it sometimes even fails with the coarse-grained setting due to problems such as availability of training data and computing resources. Loureiro et al. also gave a detailed investigation of the BERT model regarding lexical ambiguity and different semantic knowledge-based benchmarks. But they did not put much emphasis on the relationship between vector spaces and semantic knowledge. In order to better understand the emergent semantic space, Yenicelik et al. [8] investigated the vectors of polysemous words by using cluster analysis. Their study shows a similar result: BERT can to some extent distinguish different meanings of polysemous words, but with challenges that cannot be ignored. The work of Yenicelik et al. is informative about the relation between BERT embeddings and semantic knowledge, but suffers from small sample sizes (using SemCor data with approximately 500 embeddings per word) and the missing control group (monosemous words).

Unlike the above studies, the work of Garí Soler and Apidianaki [9] shows that BERT can detect the polysemy level of words as well as their sense partitionability. However, its performance is not universal. English BERT embeddings are more likely to contain polysemy-related information, but models in other languages can also distinguish between words at different polysemy levels. With carefully designed experiments, they discussed several closely related tasks: lexical polysemy detection, polysemy level prediction, the impact of frequency and POS², classification by polysemy level, and word sense cluster-

²A part-of-speech (POS) is a category of words that have similar grammatical properties. For example, noun, verb, adject-

ive. The study focuses on the macroscopic discussion of whether language models can detect word polysemy level, and does not probe deeply into the fine-grained differences within different clusters of embeddings.

Finally, how semantic clusters are formed and connected in language models has been addressed more qualitatively than quantitatively [10, 11, 6], and there are still no agreed-upon answers to these questions.

Our work differs in that we are trying to understand the geometric properties of word-specific embeddings and how they connect to semantic knowledge by conducting quantitative and qualitative analyses with the Wikicorpus.

3. Methods

In this section, we describe all the steps we follow in our analysis. Concretely, we describe the selection of target words for the analysis, the creation of contextual embeddings, the clustering of the embeddings, the computation of the *Cluster Dispersion Score (CDS)* and, finally, the summarization of each cluster.

3.1. Selection of Target Words

For the analysis described in this contribution, we selected 43 unique words (target words). The selection process reflected two requirements: 1. The selected words should have approximately the same frequency within the given corpus (to be sure that our analysis is not influenced by the frequency), 2. The selected list of words should contain examples of words with only one unique meaning (monosemous words) and words with multiple meanings (polysemous words). To satisfy the second requirement, we used the SemCor corpus [12] which is a textual corpus with each word labeled by a specific meaning from the WordNet ontology [13]. We selected 1000 words that have only one specific meaning within the SemCor corpus, and 1000 words that have more than one meaning. From these, we filtered only words with a frequency in the range of 5700–6000.

Another important criterion for selecting words is to choose words that remain the same after the tokenization process. The language model that we use for this study is XLM-RoBERTa [1], which is a transformer-based model pre-trained on a large corpus (2.5TB of filtered Common-Crawl data) in a self-supervised fashion. The model uses a tokenizer based on SentencePiece [14], and it sometimes tokenizes one word into two or more pieces. After filtering out words with this tokenization condition, we finally obtained a list of 43 target words for this study. The resulting list contains 15 words from the monosemous

ive, adverb, pronoun, preposition, etc. For more details, see https://en.wikipedia.org/wiki/Part_of_speech

category and 28 words from the polysemous category. The concrete words are listed in Table 1.

3.2. Computing the Contextual Embeddings

Our analysis of the embedding space is carried out on the *Wikicorpus* [2], which contains a large portion of the Wikipedia 2006 dump. It contains parallel contents of three languages, namely, Catalan, Spanish, and English. The size of the corpus is more than 750 million words. For our experiment, we used only English content for analysis.

To compute contextual embeddings for a given target word, we first collect all sentences from the Wikicorpus that contain this word. Each sentence is then processed by the neural language model. For our experiments, we use a transformer-based model called *XLM-RoBERTa* [1] because of its popularity in the NLP community and the available pre-trained implementation³. The model produces a vector embedding for every word within the sentence by taking other words in the sentence into account. This allows the embeddings to be contextual in contrast to *Word2Vec* [4] embeddings, which are fixed and independent of the context. We collect only the embeddings that correspond to the target word. Each embedding has a dimension of 768.

3.3. Clustering and Visualization

Our hypothesis was that distinct meanings of a given target word will form well-separated clusters in the embedding space. We wanted to detect these clusters in an unsupervised way without specifying the number of clusters in advance. For this purpose, we used the *UMAP* algorithm [15] to reduce the dimensionality of each embedding to 50 and the *HDBSCAN* clustering algorithm [16] to cluster the reduced embeddings. We set the hyperparameters of these algorithms to fixed values,⁴ but we note that for the analysis described in this paper, one could tweak the hyperparameters for each word separately. For the visualization of the clusters shown in Figure 4, we use the *UMAP* algorithm with the same hyperparameters, except that the embeddings are projected into the 2D space.

3.4. Cluster Dispersion Score

As part of our analysis, we invent a score which should measure how varied the usage of a given target word is. We call it *cluster dispersion score* or shortly *dispersion score*. It reflects the average distance between the discovered

clusters and also their size. First, we introduce a simple notation used in the definition of the score.

Let $X = \{X_1, \dots, X_n\}$ be a set of embedding vectors of a given target word and $C = \{c_1, \dots, c_m\}$ the set of indices of the clusters discovered by the clustering algorithm. We denote the distance between two clusters c_i, c_j by $d_{cl}(c_i, c_j)$ and the embeddings corresponding to the cluster c_i by $X(c_i)$. At a high level, the score has the following form:

$$CDS(X) = \sum_{c_i, c_j \in C, c_i < c_j} d_{cl}(c_i, c_j) \cdot W_{ij}.$$

It is the sum of weighted distances over all pairs of distinct clusters. If $m = 0$, the score is defined to be equal to 0. The weights and distances are symmetric; therefore, we ignore pairs with $c_i \geq c_j$. To compute the distance between two clusters, we first select the 20 most similar pairs of vectors (X_{ik}, X_{jk}) , where $X_{ik} \in X(c_i)$ and $X_{jk} \in X(c_j)$. For the similarity of two vectors, we use the cosine distance and compute it in the original 768-dimensional space. The distance between the two clusters is then the average over the 20 pairs:

$$d_{cl}(c_i, c_j) = \frac{1}{20} \sum_{k=1}^{20} d_{cos}(X_{ik}, X_{jk})$$

It is a variation of the *single linkage distance* [17], which is obtained by setting $k = 1$. Averaging over 20 most similar pairs makes the computation more robust to outliers.

The rationale behind using the closest pairs to calculate the distance instead of computing the distance between cluster centers is that sometimes the clustering algorithm splits one large cluster into multiple smaller ones as seen in Figure 4. This is not a problem if we use the closest pairs to compute the distance, because the distance will be negligible in this case and will not influence the score significantly.

The weight W_{ij} for a pair of two clusters c_i, c_j is a product of two terms:

$$W_{ij} = S_{ij} \cdot H_{ij}.$$

S_{ij} quantifies the proportion of embeddings contained in these two clusters. It is computed by:

$$S_{ij} = \frac{|X(c_i)| + |X(c_j)|}{\sum_{c_k, c_l \in C, c_k < c_l} |X(c_k)| + |X(c_l)|}.$$

The sum in the denominator normalizes the size with respect to all possible pairs. The intuition behind S_{ij} is that we want the score to be influenced more if the two clusters contain a large proportion of embeddings, compared to the case when the clusters are the same distance apart but contain only few embeddings. In the

³<https://huggingface.co/roberta-base>.

⁴For *UMAP*: `n_neighbors = 30`, `min_dist = 0.0`, and for *HDBSCAN*: `min_samples = 40`, `min_cluster_size = 50`.

second case, the clusters could correspond to a very rare usage of a given word or to outliers in the given corpus.⁵

The value of H_{ij} reflects how imbalanced the proportion of the cluster c_i is with respect to the size of the cluster c_j . This imbalance is captured by the binary entropy function H_b :

$$H_{ij} = H_b \left(\frac{|X(c_i)|}{|X(c_i)| + |X(c_j)|} \right).$$

The intuition behind H_{ij} is that we want the score to be influenced more if the two distinct clusters have approximately similar size compared to the case when one cluster contains, say, 95% and other 5% of embeddings.

3.5. Cluster summarization

In order to produce a summary of each cluster, we list 10 words with the highest *TF-IDF* score (Term Frequency – Inverse Document Frequency) [18, 19, 20]. TF-IDF is a popular score used in information retrieval that is intended to reflect how important a given word is to a document in a collection of documents. It is a product of two statistics: term frequency (how many times a given word appears in a document relative to all words in this document) and inverse document frequency (how rare is the word across all documents). In our case, we concatenate all sentences within one cluster together to form a document and then apply the TF-IDF to all clusters/documents of a given word. Before applying the TF-IDF, we remove the stop words.

4. Data and Experiments

In this section, we present the experimental results with discussion.

For this study, we selected 43 target words that contain 15 monosemous words and 28 polysemous words. For each target word, we conducted the clustering analysis based on the extracted embeddings. Then we calculated the dispersion score (Section 3.4) to measure how disperse are the clusters of a target word, see Table 1.

Comparing the dispersion scores of monosemous words and polysemous words in Figure 1 and Table 2, we can see that polysemous words have a larger mean and median. These results are in line with intuition. There should be distinct clusters of meanings for a polysemous word and the distance between these clusters should be greater than that between clusters for monosemous words. Although the polysemous word group has a larger standard deviation, it might be caused by some outliers.

For a more rigorous comparison, we ran a statistical test. We first looked at the distributions of the scores; see

⁵For example, there is a small cluster in the embeddings of the word ‘tag’ which contains only phrases ‘list by a tag’.

word	SemCor		WordNet		score
	NM	NPOS	NM	NM	
keyboard	1	1	2	2	0.0009
mystery	1	1	2	2	0.0013
buying	1	2	6	6	0.0012
conversation	1	1	1	1	0.0008
lots	1	3	11	11	0.0025
basically	1	1	1	1	0.0009
clothes	1	2	4	4	0.0006
patron	1	1	3	3	0.0016
obviously	1	1	1	1	0.0007
quest	1	2	7	7	0.0004
celebrity	1	1	2	2	0.0012
sky	1	2	2	2	0.0010
successive	1	1	1	1	0.0015
developer	1	1	2	2	0.0030
everyday	1	1	3	3	0.0015
companion	2	2	4	4	0.0015
tag	4	2	10	10	0.0036
quiet	10	4	13	13	0.0004
depression	4	1	10	10	0.0013
coin	2	2	3	3	0.0015
afternoon	2	1	2	2	0.0017
carefully	2	1	2	2	0.0010
installation	2	1	3	3	0.0011
initiative	2	2	3	3	0.0014
cruise	2	2	5	5	0.0014
export	2	2	4	4	0.0014
topic	2	1	2	2	0.0017
tight	7	2	16	16	0.0020
sheet	3	2	10	10	0.0026
girlfriend	2	1	2	2	0.0012
rap	2	2	10	10	0.0006
seal	5	2	15	15	0.0020
evident	2	1	2	2	0.0013
sweet	9	3	16	16	0.0008
span	3	2	7	7	0.0031
spin	2	2	13	13	0.0018
stem	4	2	10	10	0.0032
conductor	3	1	4	4	0.0011
employ	3	2	3	3	0.0015
configuration	2	1	2	2	0.0002
stick	6	2	25	25	0.0026
comment	4	2	6	6	0.0009
confidence	3	1	5	5	0.0012

Table 1

The overview of target words. NM: number of meanings, NPOS: number of POS. The category of monosemous words consists of words which have the value 1 in the SemCor NM column.

Figure 2. The dispersion score distributions of monosemous and polysemous words seem not to follow the normal distribution. Therefore, we applied the Rank Sum Test to see whether there were significant differences between these two groups. With the statistic = -1.4015 , p – value = 0.1611 , the statistical test shows that there are no significant differences between the dis-

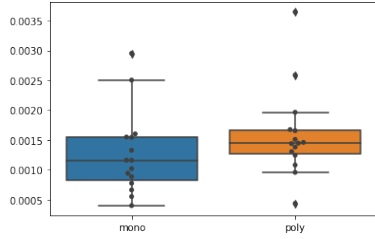


Figure 1: Boxplot of the dispersion score of monosemous and polysemous words.

descriptive statistics	mono	poly
mean	0.0013	0.0016
median	0.0012	0.0014
standard deviation	0.0007	0.0008

Table 2
Descriptive statistics of the dispersion scores of monosemous and polysemous words.

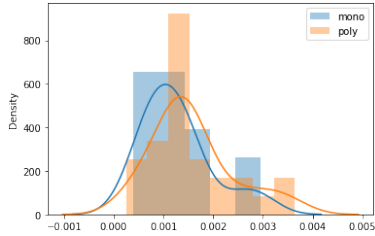


Figure 2: Distributions of cluster dispersion scores.

persion scores of monosemous and polysemous words ($p - \text{value} > 0.05$). This result contradicts our intuition and the results from descriptive statistics. Therefore, in terms of the dispersion score, we cautiously conclude that it is unclear whether there are real differences between the two groups of words. With more samples and experiments in the future, we might be able to reach a more reliable conclusion.

Furthermore, we would like to know whether there is a correlation between the dispersion score and the number of meanings a word has. Table 1 presents the number of meanings of the target words. We believe that there are two different kinds of meaning. Static meanings (in an index such as WordNet or a dictionary) and dynamic meanings (in actual texts). Table 3 and Figure 3 show that there are no strong correlations. The dispersion of clusters (representing different usages) does not correlate with the number of meanings (and POS) a word has. Word A, for example, may only have two meanings while word

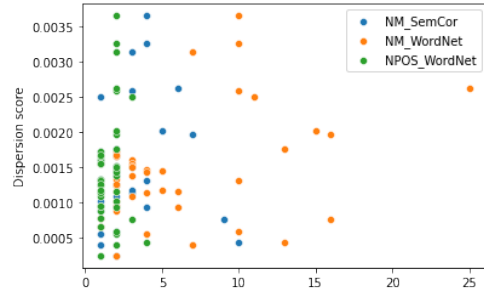


Figure 3: The scatter plot of the dispersion scores.

B may have ten. However, the cluster distances of word A may be greater than that of word B. The reasons may be because word A has two very distinct meanings and contexts, whereas word B has ten meanings and contexts that are more similar. A closer look at the clusters will help us understand the factors that influence dispersion scores.

	NM_SemCor	NM_WordNet	NPOS_WordNet
DS	0.1371	0.3924	0.1499

Table 3
The correlation coefficient. DS: dispersion score, NM: number of meanings, NPOS: number of POS.

4.1. Closer Look at the Selected Words

Looking at the monosemous words in Table 1 (those having the value 1 in the SemCor NM column), we can see that there are two outliers (“lots” and “developer”) that have the dispersion score much higher than other words in this category. In Figure 4, we show the UMAP visualization of these two words together with two words from the polysemous category (“stick” and “sheet”). The clusters are colored according to the labels assigned by the clustering algorithm. Next to each cluster, we display 10 words (or 5 for the word “stick”) with the highest TF-IDF score. As can be seen in the plot for the word “lots”, there are three distinct clusters. Two of them larger and one smaller. The two larger clusters correspond to the following meanings: lots as “parcels of land” and lots as in “lots of people, money, etc.” and the smaller cluster contains sentences with “parking lots”. Clusters in the other three plots can be interpreted in a similar way.

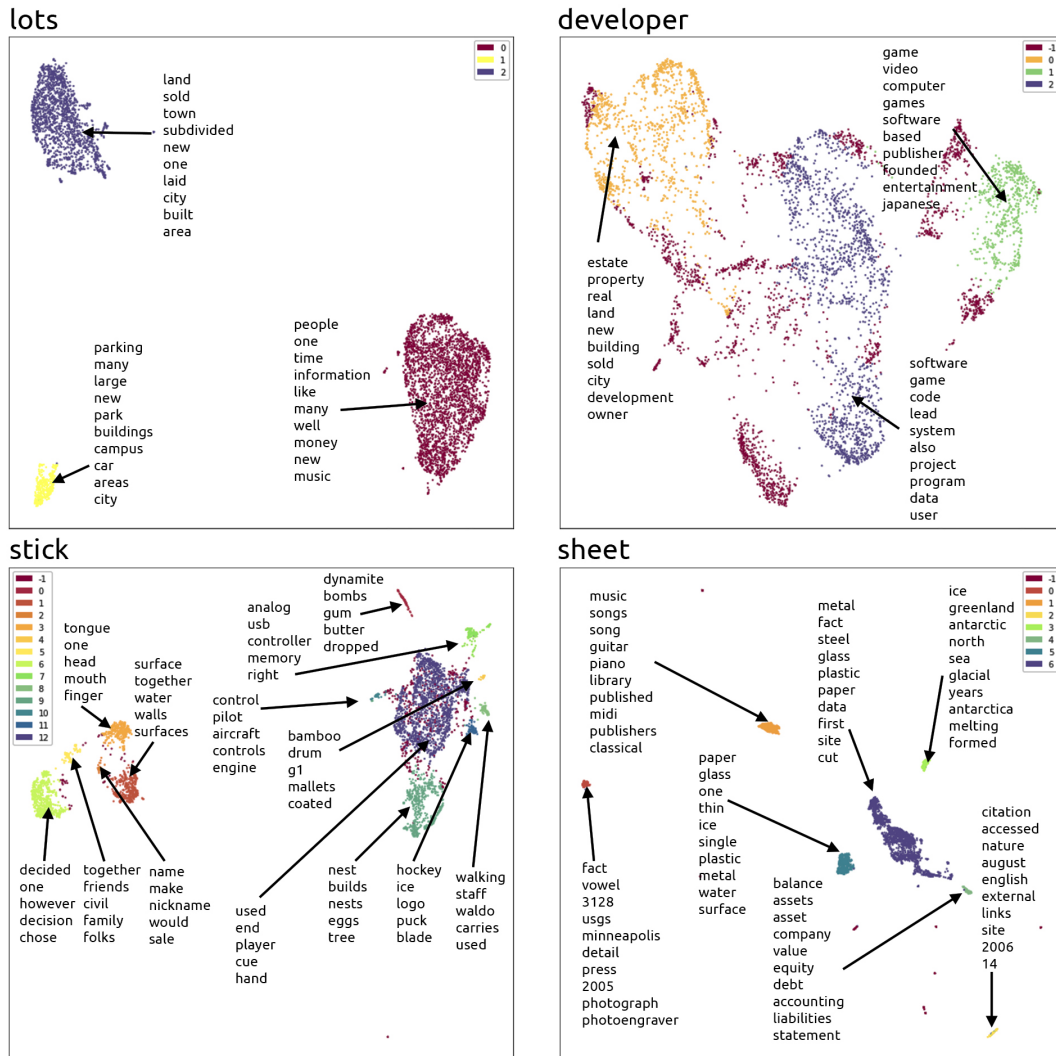


Figure 4: This figure shows a UMAP visualization of embeddings of four selected words. The embeddings are colored according to the class assigned by the clustering algorithm. The dark red color corresponds to the cluster '-1', which contains outliers. The clustering was done in 50-dimensional space and therefore the 2D visualization may distort the geometry used for the clustering. Next to each cluster, we display 10 (or 5 in the case of the word 'stick') words with the highest TF-IDF score.

5. Discussion

After taking a closer look at the discovered clusters of each word, we can see that it is not clear when to distinguish one meaning as separate from the other. For example, for the word *developer*, there is a well-separated cluster corresponding to the sentences containing the phrase “game developer” and another cluster corresponding to sentences about software developers. Similar nuances can also be seen in several other words. This observation questions the completeness of manually defined lists of

word meanings, such as those given by WordNet and other sources. One could also realize that the clusters are largely determined by the given corpus, which is a small snapshot of the language used at a specific time and place. It reflects distinctions that are important to the people who wrote the texts contained in the corpus. Such distinctions arise because of real needs of the people using the language (e.g., Inuits having a large number of distinct words for different types of snow). As can be seen in Figure 4, neural language models can discover these distinctions just by learning to predict a word from

its context.

We also mention a few problematic points in our method. The most problematic point is that the dispersion score is unstable with respect to larger changes of hyperparameters of the clustering algorithm. We tried to design the score to be stable with respect to splits of larger clusters into multiple smaller ones, but more work would need to be done in order to really achieve this stability.

Next, as discovered by Timkey et al. [21], the similarity of embeddings created by transformer-based language models may be greatly influenced by very few dimensions of the embedding. These dimensions apparently distort the cosine similarity and disable distinguishing nuanced meanings. Timkey et al. suggest to normalize the embeddings before measuring the cosine similarity as a simple way to mitigate this problem. In our experiments, we have not seen this problem, as the clusters were often well separated, but we plan to use the proposed normalization in the future.

Lastly, the range of selected words is very limited due to the requirement of similar frequency and no subword tokenization, as mentioned in Section 3.1. In the future, we plan to conduct a more extensive analysis without these limitations.

6. Conclusion

In this contribution, we provided a quantitative and qualitative analysis of the semantic vector space induced by a neural language model and a corpus. We showed that the contextual embeddings created by the language model often form well-separated clusters that correspond to different meanings of the word. As part of our analysis, we introduced a score that reflects how dispersed is the collection of clusters for a given word. Our analysis shows that the score is not directly correlated with the number of meanings as defined by WordNet. After closer inspection of several words, we concluded that it is not clear when one meaning should be separated from the other and that manually defined lists of different meanings of the word are not complete or fine-grained enough. Our analysis also shows the possibility of developing applications that will create a list of different usages of the word in an automatic data-driven way. We envision that such applications may be useful for foreign language learners.

References

- [1] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, V. Stoyanov, Unsupervised cross-lingual representation learning at scale, *CoRR* abs/1911.02116 (2019). URL: <http://arxiv.org/abs/1911.02116>. arXiv:1911.02116.
- [2] S. Reese, G. Boleda, M. Cuadros, L. Padró, G. Rigau, Wikicorpus: A word-sense disambiguated multilingual Wikipedia corpus, in: *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, European Language Resources Association (ELRA), Valletta, Malta, 2010. URL: http://www.lrec-conf.org/proceedings/lrec2010/pdf/222_Paper.pdf.
- [3] M. T. Pilehvar, J. Camacho-Collados, WiC: the word-in-context dataset for evaluating context-sensitive meaning representations, in: *Proceedings of NAACL-HLT*, 2019, pp. 1267–1273.
- [4] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, *arXiv preprint arXiv:1301.3781* (2013).
- [5] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, *Advances in Neural Information Processing Systems* 30 (2017).
- [6] G. Wiedemann, S. Remus, A. Chawla, C. Biemann, Does BERT make any sense? Interpretable word sense disambiguation with contextualized embeddings, *arXiv preprint arXiv:1909.10430* (2019).
- [7] D. Loureiro, K. Rezaee, M. T. Pilehvar, J. Camacho-Collados, Analysis and evaluation of language models for word sense disambiguation, *Computational Linguistics* 47 (2021) 387–443.
- [8] D. Yenicelik, F. Schmidt, Y. Kilcher, How does BERT capture semantics? A closer look at polysemous words, in: *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, 2020, pp. 156–162.
- [9] A. Garí Soler, M. Apidianaki, Let’s play mono-poly: BERT can reveal words’ polysemy level and partitionability into senses, *Transactions of the Association for Computational Linguistics* 9 (2021) 825–844.
- [10] E. Reif, A. Yuan, M. Wattenberg, F. B. Viegas, A. Coenen, A. Pearce, B. Kim, Visualizing and measuring the geometry of BERT, *Advances in Neural Information Processing Systems* 32 (2019).
- [11] G. Jawahar, B. Sagot, D. Seddah, What does BERT learn about the structure of language?, in: *ACL 2019-57th Annual Meeting of the Association for Computational Linguistics*, 2019.
- [12] G. A. Miller, C. Leacock, R. Teng, R. T. Bunker, A semantic concordance, in: *Human Language Technology: Proceedings of a Workshop Held at Plainsboro, New Jersey, March 21-24, 1993*, 1993, pp. 303–308.
- [13] G. A. Miller, WordNet: a lexical database for English, *Communications of the ACM* 38 (1995) 39–41.
- [14] T. Kudo, J. Richardson, SentencePiece: A simple and

- language independent subword tokenizer and detokenizer for Neural Text Processing, arXiv preprint arXiv:1808.06226 (2018).
- [15] L. McInnes, J. Healy, J. Melville, UMAP: Uniform manifold approximation and projection for dimension reduction, arXiv preprint arXiv:1802.03426 (2018).
 - [16] R. J. Campello, D. Moulavi, J. Sander, Density-based clustering based on hierarchical density estimates, in: Pacific-Asia Conference on Knowledge Discovery and Data Mining, Springer, 2013, pp. 160–172.
 - [17] R. Sibson, SLINK: an optimally efficient algorithm for the single-link cluster method, *The Computer Journal* 16 (1973) 30–34.
 - [18] A. Rajaraman, J. D. Ullman, *Mining of Massive Datasets*, Cambridge University Press, 2011.
 - [19] K. S. Jones, A statistical interpretation of term specificity and its application in retrieval, *Journal of Documentation* (1972).
 - [20] H. P. Luhn, A statistical approach to mechanized encoding and searching of literary information, *IBM Journal of Research and Development* 1 (1957) 309–317.
 - [21] W. Timkey, M. van Schijndel, All bark and no bite: Rogue dimensions in transformer language models obscure representational quality, arXiv preprint arXiv:2109.04404 (2021).