How Contextualized Word Embeddings Represent Word Senses

Rocco Tripodi
University of Bologna
rocco.tripodi@unibo.it

Abstract

English. Contextualized embedding models, such as ELMo and BERT, allow the construction of vector representations of lexical items that adapt to the context in which words appear. It was demonstrated that the upper layers of these models capture semantic information. This evidence paved the way for the development of sense representations based on words in context. In this paper, we analyze the vector spaces produced by 11 pre-trained models and evaluate these representations on two tasks. The analysis shows that all these representations contain redundant information. The results show the disadvantage of this aspect.

Italiano. *Modelli come ELMo o BERT consentono di ottenere rappresentazioni vettoriali delle parole che si adattano al contesto in cui queste appaiono. Il fatto che i livelli alti di questi modelli immagazzinino informazione semantica ha portato a sviluppare rappresentazioni di senso basate su parole nel contesto. In questo lavoro analizziamo gli spazi vettoriali prodotti con 11 modelli pre-addestrati e valutiamo le loro prestazioni nel rappresentare i diversi sensi delle parole. Le analisi condotte mostrano che questi modelli contengono informazioni ridondanti. I risultati evidenziano le criticità inerenti a questo aspetto.*

1 Introduction

The introduction of contextualized embedding models, such as ELMo (Peters et al., 2018) and BERT (Devlin et al., 2019), allows the construction of vector representations of lexical items that adapt to the context in which words appear. It has been shown that the upper layers of these models contain semantic information (Jawahar et al., 2019) and are more diversified than lower layers (Ethayarajh, 2019). These word representations overcame the meaning conflation deficiency that affects static word embedding techniques (Camacho-Collados and Pilehvar, 2018; Tripodi and Pira, 2017), such as *word2vec* (Mikolov et al., 2013) or *GloVe* (Pennington et al., 2014) thanks to the adaptation to the context of use.

The evaluation of these models has been conducted mainly on downstream tasks (Wang et al., 2018; Wang et al., 2019). With extrinsic evaluations, the models are fine-tuned, adapting the vector representations to specific tasks. The resulting vectors are then used as features in classification problems. This hinders a direct evaluation and analysis of the models because the evaluation also takes into account the ability of the classifier to learn the task. A model trained for this kind of task may learn only to discriminate among features that belong to each class with poor generalization.

The interpretability of neural networks is an emerging line of research NLP that aims at analyzing the properties of pre-trained language models (Belinkov and Glass, 2019). Different studies have been conducted in recent years to discover what kind of linguistic information is stored in large neural language models. Many of them are focused on syntax (Hewitt and Manning, 2019; Jawahar et al., 2019) and attention (Michel et al., 2019; Kovaleva et al., 2019). For what concerns semantics, the majority of the studies focus on common knowledge (Petroni et al., 2019) and inference and role-based event prediction (Ettinger, 2020). Only a few of them have been devoted to lexical semantics, for example, Reif et al. (2019) show how different representations of the

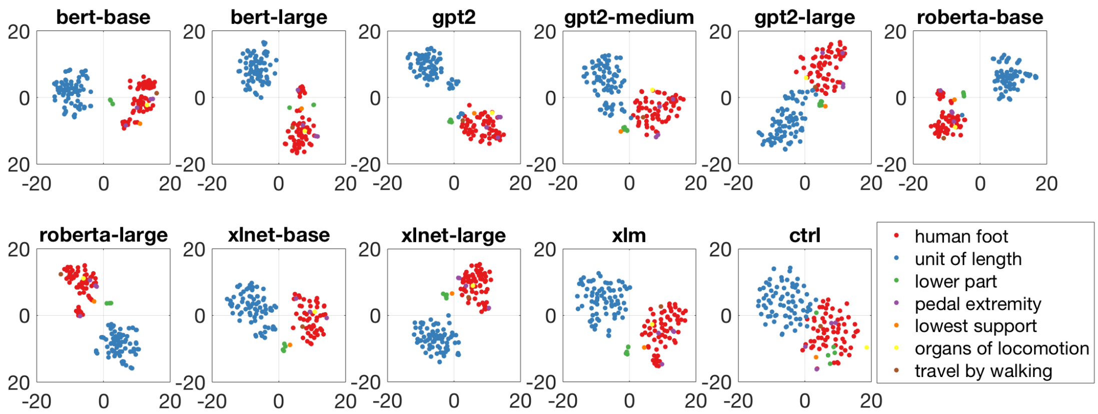

Figure 1: t-SNE representations for the word *foot* in SemCor, grouped by sense.

same lexical form tend to cluster according to their sense.

In this work, we propose an in-depth analysis of the properties of the vector spaces induced by different embedding models and an evaluation of their word representations. We present how the properties of the vector space contribute to the success of the models in two tasks: sense induction and word sense disambiguation. In fact, even if contextualized models do not create one representation per word sense (Ethayarajh, 2019), their contextualization create similar representations for the same word sense that can be easily clustered.

2 Related Work

Given the success (and the opacity) of contextualized embedding models, many works have been proposed to analyze their inner representations. These analyses are based on *probing tasks* (Conneau et al., 2018) that aim at measuring how the information extracted from a pre-trained model is useful to represent linguistic structures. Probing tasks involve training a diagnostic classifier to determine if it encodes desired features. Tenney et al. (2019) discovered that specific BERT's layers are more suited for representing information useful to solve specific tasks and that the ordering of its layers resembles the ordering of a traditional NLP pipeline: POS tagging, parsing, NER, semantic role labeling, and coreference resolution. Hewitt and Manning (2019) evaluated whether syntax trees are embedded in a linear transformation of a neural network's word representation space. Hewitt and Liang (2019) raised the problem of interpreting the results derived from probing analysis. In fact, it is difficult to understand whether high accuracy values are due to the representation itself or, instead, they are the result of the ability to learn a specific task during training.

Our work is more in line with works that try to find general properties of the representations generated by different contextualized models. For example, Mimno and Thompson (2017) demonstrated that the vector space produced by a static embedding model is concentrated in a narrow cone and that its concentration depends on the ratio of positive and negative examples. Mu and Viswanath (2018) explored this analysis further, demonstrating that the embedding vectors share the same common vector and have the same main direction. Ethayarajh (2019) demonstrated how upper layers of a contextualizing model produce more contextualized representations. We built on top of these works analyzing the vector space generated by contextualized models and evaluating them.

3 Construction of the Vector Spaces

We used SemCor (Miller et al., 1993) as reference corpus for our work. This choice is motivated by the fact that it is the largest dataset manually annotated with sense information and it is commonly used as training set for word sense disambiguation. It contains 352 documents whose content words (about $226,000$) have been annotated with WordNet (Miller, 1995) senses. In total there are $33,341$ unique senses distributed over $22,417$ different words. The sense distribution in this corpus is very skewed, and follows a power law (Kilgarriff, 2004). This makes the identification of senses challenging. The dataset is also difficult due to the

Model	training data	vocab. size	n. param.	vec. dim.	objective
$\text{BERT}_{\text{base}}$ (Devlin et al., 2019)	16GB	30K	110M	768	masked language model and next sentence prediction
$\text{BERT}_{\text{large}}$ (Devlin et al., 2019)	16GB	30K	340M	1024	masked language model and next sentence prediction
$\text{GPT-2}_{\text{base}}$ (Radford et al., 2019)	40GB	50K	117M	768	language model
$\text{GPT-2}_{\text{medium}}$ (Radford et al., 2019)	40GB	50K	345M	1024	language model
$\text{GPT-2}_{\text{large}}$ (Radford et al., 2019)	40GB	50K	774M	1280	language model
$\text{RoBERTa}_{\text{base}}$ (Liu et al., 2019)	160GB	50K	125M	768	masked language model
$\text{RoBERTa}_{\text{large}}$ (Liu et al., 2019)	160GB	50K	355M	1024	masked language model
$\text{XLNet}_{\text{base}}$ (Yang et al., 2019)	126GB	32K	110M	768	bidirectional language model
$\text{XLNet}_{\text{large}}$ (Yang et al., 2019)	126GB	32K	340M	1024	bidirectional language model
$\text{XLM}_{\text{english}}$	16GB	30K	665M	2048	language model
CTRL (Keskar et al., 2019)	140GB	250K	1.63B	1280	conditional transformer language model

Table 1: Statistics and hyperparameters of the models.

Model	$\mathbf{AvgNorm}$	$\mathbf{MeanVecNorm(A)}$	$\mathbf{MeanVecNorm(\hat{A})}$	$\mathbf{avg.MEV}$	$\mathbf{avg.IntSim}$	$\mathbf{avg.ExtSim}$
$\text{BERT}_{\text{base}}$	25.78 ± 1.28	17.94	17.84	0.43 ± 0.18	0.74 ± 0.05	0.69 ± 0.06
$\text{BERT}_{\text{large}}$	20.83 ± 2.51	12.43	11.58	0.38 ± 0.18	0.66 ± 0.08	0.59 ± 0.08
$\text{GPT-2}_{\text{base}}$	125.13 ± 10.25	91.46	90.99	0.46 ± 0.18	0.79 ± 0.05	0.76 ± 0.05
$\text{GPT-2}_{\text{medium}}$	427.45 ± 38.78	371.86	360.36	0.51 ± 0.18	0.85 ± 0.03	0.84 ± 0.03
$\text{GPT-2}_{\text{large}}$	290.29 ± 38.56	226.39	212.97	0.43 ± 0.18	0.75 ± 0.05	0.72 ± 0.05
$\text{RoBERTa}_{\text{base}}$	25.78 ± 0.56	22.17	22.25	0.51 ± 0.17	0.87 ± 0.02	0.85 ± 0.03
$\text{RoBERTa}_{\text{large}}$	31.47 ± 0.65	26.99	27.04	0.52 ± 0.18	0.88 ± 0.02	0.84 ± 0.03
$\text{XLNet}_{\text{base}}$	47.68 ± 0.66	43.28	43.26	0.53 ± 0.17	0.88 ± 0.01	0.87 ± 0.02
$\text{XLNet}_{\text{large}}$	28.27 ± 1.42	19.56	19.68	0.38 ± 0.17	0.66 ± 0.04	0.62 ± 0.05
$\text{XLM}_{\text{english}}$	44.92 ± 2.61	37.13	36.7	0.45 ± 0.18	0.79 ± 0.03	0.77 ± 0.03
CTRL	4443.62 ± 351.98	3927.86	3879.56	0.49 ± 0.18	0.84 ± 0.02	0.83 ± 0.02

Table 2: Detailed description of the embedding space produced with each model.

fine granularity of WordNet (Navigli, 2006).

To construct the vector space A from SemCor we collected all the senses S_i of a word w_i and for each sense $s_j \in S_i$ we recovered the sentences $\{Sent_1^{w_i s_j}, Sent_2^{w_i s_j}, ..., Sent_n^{w_i s_j}\}$ in which this particular sense occurs. These sentences are then fed into a pre-trained model and the token embedding representations of word w_i, $\{e_1^{w_i s_j}, e_2^{w_i s_j}, ..., e_n^{w_i s_j}\}$, are extracted from the last hidden layer. This operation is repeated for all the senses in S_i, and for all the tagged words in the vocabulary, V. The vector space corresponds to all the representations of the words in V.

A t-SNE visualization of the different embeddings in SemCor for the word *foot* is presented in Figure 1. In this Figure, we can see that the three main senses of *foot* (i.e., human foot, unit of length and lower part) occupy a definite position in the vector space, suggesting that the models are able to produce specific representations for the different senses of a word and that they lie on defined subspaces. In this work we want to test to what extent this feature is present in language models.

Implementations details The pre-trained models used in this study are: two BERT (Devlin et al., 2019) models, *base cased* (12-layer, 768-hidden, 12-heads, 110M parameters) and *large cased* (24-layer, 1024-hidden, 16-heads, 340M parameters); three GPT-2 (Radford et al., 2019) models, *base* (12-layer, 768-hidden, 12-heads, 117M parameters), *medium* (24-layer, 1024-hidden, 16-heads, 345M parameters) and *large* (36-layer, 1280-hidden, 20-heads, 774M parameters); two RoBERTa (Liu et al., 2019) models, *base* (12-layer, 768-hidden, 12-heads, 125M parameters) and *large* (24-layer, 1024-hidden, 16-heads, 355M parameters); two XLNet (Yang et al., 2019) models, *base* (12-layer, 768-hidden, 12-heads, 110M parameters) and *large* (24-layer, 1024-hidden, 16-heads, 340M parameters); one XLM (Lample et al., 2019) model (12-layer, 2048-hidden, 16-heads) and one CTRL (Keskar et al., 2019) model (48-layer, 1280-hidden, 16-heads, $1.6B$ parameters). The main features of these models are summarized in Table 1. We averaged the embeddings of sub-tokens to obtain token-level representations.

3.1 Analysis

The first objective of this work is to analyze the vector space produced with the models. This analysis is aimed at investigating the properties of the contextualized vectors. A detailed description of the embedding spaces constructed with the pre-

We used the transformers library (Wolf et al., 2019).

trained models is presented in Table 2. We computed the norm for all the vectors in the vector space A, and averaged them:

$$AvgNorm = \frac{1}{|A|}\sum_{i=1}^{|A|} \|e_i\|_2. \quad (1)$$

This measure gives us an intuition on how diverse the semantic space constructed with the different models is. In fact, we can see that the magnitude of the vectors constructed with BERT, RoBERTa, XLNet, and XLM is low while those of GPT-2 and CTRL are very high.

We computed also the norm of the vector resulting in averaging all the vectors in the semantic space V, as:

$$MeanVecNorm = \left\| \frac{1}{|A|}\sum_{i=1}^{|A|} e_i \right\|_2. \quad (2)$$

All the semantic spaces have non-zero mean and the mean norm is high. This result suggests that the vectors contain redundant information and share a common nonzero vector. This is not only because the vector space contains representations of the same sense. In fact, if we create a new semantic space, $\hat{A}$, averaging all the representations of the same word sense, the $MeanVecNorm$ of this space is still high for all the models.

We used the Maximum Explainable Variance (MEV) for the representations of each word in V. This measure corresponds to the proportion of the variance in the embeddings that can be explained by their first principal components and was computed as:

$$MEV(w) = \frac{\sigma_1^2}{\sum_i \sigma_i^2}. \quad (3)$$

where $\sigma_i^2 1$ is the first principal component of the vector space A. It can give an upper bound on how contextualized representations can be replaced by a static embedding (Ethayarajh, 2019). The model with the lowest MEV is $\text{BERT}_{\text{large}}$ and $\text{XLNet}_{\text{large}}$.

The other measures that we used for the evaluation of the vector space are based on the very notion of a cluster, which imposes that the data points inside a cluster must satisfy two conditions: internal similarity and external dissimilarity (Pelillo, 2009). To this end, we used the senses of each word in the vocabulary of SemCor as clusters and extracted the corresponding vectors from V. We

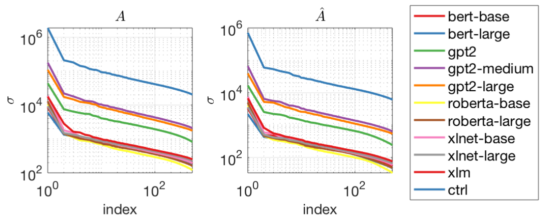

Figure 2: The first 500 principal components computed on A and $\hat{A}$.

then computed the *internal similarity* of a cluster, c, as:

$$IntSim(c) = \frac{1}{n^2 - n}\sum_{j}\sum_{k \neq j} cos(e_j, e_k), \quad (4)$$

where n is the number of data points in the cluster. We computed also the *external similarity* of a cluster c by computing the cosine similarity among each point in c and all the points in the subspace S induced by the senses of a word that has c as one of its senses:

$$ExtSim(c) = \frac{1}{n \cdot m}\sum_{j=1}^{n}\sum_{k=1}^{m} cos(e_j, e_k), \quad (5)$$

where m is the total number of data points in the subspace S (excluding those in c) and n is the number of points in the cluster c. Our hypothesis is that good representations should have high internal similarity and low external similarity and that the difference between them should be high.

As it can be seen from Table 2 the internal similarity is higher than the external for all the models. Despite this, the scores are in a wide range. The lowest $IntSim$ is given by $\text{BERT}_{\text{large}}$ and the highest by $\text{RoBERTa}_{\text{large}}$ and $\text{XLNet}_{\text{base}}$. The lowest $ExtSim$ is given by $\text{BERT}_{\text{large}}$ and the highest by $\text{XLNet}_{\text{base}}$. The largest difference between the two measures is given by $\text{BERT}_{\text{large}}$. $\text{RoBERTa}_{\text{large}}$ gives has also a large gap between the two measures, furthermore, their standard deviation is very low. As we will see in Section 4 these last two models perform better than others in clustering and classification tasks.

4 Evaluation

Sense Induction This task is aimed at understanding if representations belonging to different senses can be separated using an unsupervised approach. We hypothesize that a good contextualization process should produce more discriminative

model	k-means					dominant-set				
	N	V	A	R	All	N	V	A	R	All
$\text{BERT}_{\text{base}}$	57.2	50.6	56.2	62.0	54.9 ± 14.8	55.7	45.3	51.7	45.8	51.0 ± 17.5
$\text{BERT}_{\text{large}}$	**59.3**	51.9	56.9	59.0	56.2 ± 15.3	53.4	42.6	46.8	39.9	47.8 ± 17.1
$\text{GPT-2}_{\text{base}}$	54.1	48.3	55.6	56.8	<u>52.3</u>±14.7	54.3	45.3	50.2	46.3	50.1 ± 17.2
$\text{GPT-2}_{\text{medium}}$	53.9	49.1	56.2	59.8	52.8 ± 14.5	59.7	49.8	58.7	54.8	56.0 ± 18.8
$\text{GPT-2}_{\text{large}}$	53.8	49.4	58.1	58.8	53.0 ± 14.8	50.2	44.1	46.1	44.1	47.1 ± 16.0
$\text{RoBERTa}_{\text{base}}$	56.4	51.4	56.7	59.7	54.8 ± 14.7	65.3	55.1	64.8	61.4	61.6 ± 19.2
$\text{RoBERTa}_{\text{large}}$	58.5	**53.0**	**58.6**	**62.7**	**56.7**±14.9	66.7	**56.6**	66.3	**64.2**	**63.2**±19.3
$\text{XLNet}_{\text{base}}$	54.2	49.1	53.8	56.8	52.2 ± 14.4	**67.2**	55.0	**68.7**	63.8	<u>62.7</u>±20.7
$\text{XLNet}_{\text{large}}$	57.6	52.5	57.9	60.8	<u>55.9</u>±14.4	51.0	44.8	47.5	40.9	<u>47.6</u>±15.0
$\text{XLM}_{\text{english}}$	56.3	50.1	56.5	62.1	54.3 ± 15.1	60.4	51.3	59.5	55.9	57.0 ± 18.1
CTRL	53.8	47.0	56.5	57.4	51.9 ± 15.4	60.4	49.4	61.7	56.3	56.8 ± 19.2

Table 3: Results (as average accuracy) on clustering divided by algorithm and part of speech: nouns (N), verbs (V), adjectives (A), adverbs (R) and on the concatenations of all datasets (All).

representations that can be easily identified by a clustering algorithm.

We used the sense clusters extracted from SemCor as ground truth for this experiment (see Section 3) and grouped them if they are senses of the same word (with a given part of speech). We retained only the groups that have at least 20 data points and we discarded also monosemous words for the evaluation on k-means. The resulting datasets consist of 1871 (entire) and 1499 (without monosemous words) sub-datasets with $141,074$ and $116,019$ data points in total, respectively. We computed the accuracy on each sub-dataset computing the number of data points that have been clustered correctly and averaged the results to measure the performance of each model.

The first algorithm is k-means (Lloyd, 1982). It is a partitioning, iterative algorithm whose objective is to minimize the sum of point-to-centroid distances, summed over all k clusters. We used the k-means++ heuristic (Arthur and Vassilvitskii, 2007) and the cosine distance metric to determine distances. We selected this algorithm because it is simple, non-parametric, and is widely used. It is important to notice that k-means requires the number of clusters to extract, for this reason, we restricted the evaluation only to ambiguous words.

The second algorithm used is *dominant-set* (Pavan and Pelillo, 2007). It is a graph-based algorithm that extracts compact structures from graphs generalizing the notion of maximal clique defined on unweighted graphs to edge-weighted graphs. We selected this algorithm because it is non-parametric, requires only the adjacency matrix of a weighted graph as input, and, more importantly, does not require the number of clusters to extract. The clusters are extracted from the graph sequentially using a peel-off strategy. This feature allows us to include in the evaluation also unambiguous words and to see if their representations are grouped into a single cluster or partitioned into different ones. We used cosine similarity to weigh the edges of the input graph.

The results of this evaluation are presented in Table 3. RoBERTa and BERT have the overall best performances on this task using both algorithms. In particular, $\text{RoBERTa}_{\text{large}}$ performs consistently well on all parts of speech and across algorithms, while other models perform well only in combination with one of the two algorithms. This is presumably owing to the big gap between the internal and the external similarity produced by this model, as explained in Section 3.1.

This evaluation tends to confirm the claim that larger versions of the same model achieve better results. From Table 3, we can also see that the models have more difficulties in identifying the different senses of verbs, while nouns and adverbs have higher results. This is probably due to the different distribution of these word classes in the training sets of the models and WordNet's fine-granularity. The performances of the models with dominant-set are surprisingly high, considering that the setting of this experiment is completely unsupervised. Furthermore, this algorithm is conceived to extract compact clusters and this feature could drive it to over partition the vector space of monosemous words. Instead, the results suggest the opposite: that the models are able to produce representations with high internal similarity, positioning their representations on a defined sub-space.

Word Sense Disambiguation We used the method proposed in Peters et al. (2018) to create

Model	S2			S3			SE07			SE13			SE15			All		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
$\text{BERT}_{\text{base}}$	80.6	67.9	73.7	77.2	68.8	72.8	66.4	63.1	64.7	74.4	62.7	68.1	78.3	68.8	73.2	77.0	66.8	71.5
$\text{BERT}_{\text{large}}$	81.2	68.4	74.3	80.3	71.5	**75.6**	68.5	65.1	66.7	75.8	63.9	69.3	79.7	70.1	74.6	77.9	67.5	72.3
$\text{GPT-2}_{\text{base}}$	75.6	63.7	69.1	71.5	63.7	67.4	59.3	56.3	57.7	71.8	60.5	65.7	74.4	65.4	69.6	72.4	62.8	67.2
$\text{GPT-2}_{\text{medium}}$	76.5	64.5	70.0	72.9	65.0	68.7	62.0	58.9	60.4	74.0	62.3	67.7	76.6	67.3	71.7	74.0	64.2	68.8
$\text{GPT-2}_{\text{large}}$	76.4	64.4	69.9	72.1	64.2	67.9	61.8	58.7	60.2	72.8	61.4	66.6	75.6	66.3	70.7	73.4	63.6	68.1
$\text{RoBERTa}_{\text{base}}$	82.0	69.1	**75.0**	79.4	70.7	74.8	66.7	63.3	64.9	75.5	63.7	69.1	79.5	69.9	74.4	78.5	68.0	72.9
$\text{RoBERTa}_{\text{large}}$	82.0	69.1	**75.0**	80.0	71.2	75.4	70.6	67.0	**68.8**	77.1	65.0	**70.5**	81.0	71.1	**75.7**	79.4	68.9	**73.8**
$\text{XLNet}_{\text{base}}$	78.8	65.8	71.7	76.2	67.4	71.5	67.3	63.7	65.5	70.7	58.3	63.9	77.5	67.1	71.9	75.4	64.6	69.5
$\text{XLNet}_{\text{large}}$	80.6	67.9	73.7	78.7	70.1	74.2	67.6	64.2	65.8	75.3	63.5	68.9	80.6	70.8	75.4	78.0	67.7	72.5
CTRL	73.4	61.9	67.1	70.1	62.5	66.1	54.2	51.4	52.8	68.2	57.5	62.4	72.3	63.5	67.6	69.9	60.6	64.9

Table 4: Results indicating precision (P), recall (R) and F1 on each dataset and on their concatenation (All). All the results are computed using $\hat{A}$ as vector space.

sense vectors from contextualized word vectors. This method consists in averaging all the representations of a given sense. The resulting vector space corresponds to $\hat{A}$ (see Section 3.1). We evaluated the generated vectors on a standard benchmark (Raganato et al., 2017) for WSD. It consists of five datasets that were unified to the same WordNet version: Senseval-2 (S2), Senseval-3 (S3), SemEval-2007 (S7), SemEval-2013 and SemEval-2015, having in total $10,619$ target words.

The identification of word senses is conducted by feeding the entire texts of the datasets into a pre-trained model and extracting, for each target word w_i, its embedding representation $e_k^{w_i}$ as was done for the construction of the semantic space. Once these representations are available, we compute the cosine similarities among $e_k^{w_i}$ and the embeddings in $\hat{A}$ constructed with the same model and selected the sense with the highest similarity. We did not use more sophisticated models such as WSD-games (Tripodi and Navigli, 2019; Tripodi et al., 2016) because we wanted to keep the evaluation as simple as possible as not to influence the evaluation of the results.

The results of this evaluation are presented in Table 4. The first trend that emerges from the results is the big gap between *precision* and *recall*. This is due to the absence of many senses in our training set. We did not want to use back-off strategies or other techniques usually employed in the WSD literature, to not influence the performances and the analysis of the results. Despite the simplicity of the approach, it performs surprisingly well. In particular, BERT, RoBERTa, and XLNet (three bidirectional models) have very high results. The low performances of CTRL are probably due to its large vocabulary and to its objective, designed to solve different tasks.

5 Conclusion and Future Work

We conducted an extensive analysis of the semantic capabilities of contextualized embedding models. We analyzed the vector space constructed using pre-trained models and found that their vectors contain redundant information and that their first two principal components are dominant.

The results on sense induction are promising. They demonstrated the effectiveness of contextualized embeddings to capture semantic information. We did not find higher performances from more complex models, rather, we found that RoBERTa, a model that was developed by simplifying a more complex model, BERT, was one of the best performers. Neither the dimension of the hidden layers, the size of the training data, nor the size of the vocabulary seems to play a big role in modeling semantics. As stated in previous works, inserting an anisotropy penalty to the objective function of the models could improve directly the representations. We also noticed that, even if BERT models and XLNet have different objectives and are trained on different data, they have similar performances. It emerged that these models are less redundant than others.

The conclusion that we can draw from our analysis and evaluation is that pre-trained language models can capture lexical-semantic information and that unsupervised models can be used to distinguish among their representations. On the other hand, these representations are redundant and anisotropic. We hypothesize that reducing these aspects can lead to better representations. This operation can be carried out *post-hoc* but we think that training new models keeping this point in mind could lead to the development of better models.

References

David Arthur and Sergei Vassilvitskii. 2007. k-means++: the advantages of careful seeding. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2007, New Orleans, Louisiana, USA, January 7-9, 2007*, pages 1027–1035.

Yonatan Belinkov and James Glass. 2019. Analysis methods in neural language processing: A survey. *Transactions of the Association for Computational Linguistics*, 7:49–72, March.

José Camacho-Collados and Mohammad Taher Pilehvar. 2018. From word to sense embeddings: A survey on vector representations of meaning. *J. Artif. Intell. Res.*, 63:743–788.

Alexis Conneau, German Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. 2018. What you can cram into a single $&!#* vector: Probing sentence embeddings for linguistic properties. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2126–2136, Melbourne, Australia, July. Association for Computational Linguistics.

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.

Kawin Ethayarajh. 2019. How contextual are contextualized word representations? comparing the geometry of BERT, ELMo, and GPT-2 embeddings. 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 55–65, Hong Kong, China, November. Association for Computational Linguistics.

Allyson Ettinger. 2020. What bert is not: Lessons from a new suite of psycholinguistic diagnostics for language models. *Transactions of the Association for Computational Linguistics*, 8:34–48.

John Hewitt and Percy Liang. 2019. Designing and interpreting probes with control tasks. 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 2733–2743, Hong Kong, China, November. Association for Computational Linguistics.

John Hewitt and Christopher D. Manning. 2019. A structural probe for finding syntax in word representations. 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 4129–4138, Minneapolis, Minnesota, June. Association for Computational Linguistics.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What does BERT learn about the structure of language? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3651–3657, Florence, Italy, July. Association for Computational Linguistics.

Nitish Shirish Keskar, Bryan McCann, Lav R Varshney, Caiming Xiong, and Richard Socher. 2019. Ctrl: A conditional transformer language model for controllable generation. *arXiv preprint arXiv:1909.05858*.

Adam Kilgarriff. 2004. How dominant is the commonest sense of a word? In Petr Sojka, Ivan Kopeček, and Karel Pala, editors, *Text, Speech and Dialogue*, pages 103–111, Berlin, Heidelberg. Springer Berlin Heidelberg.

Olga Kovaleva, Alexey Romanov, Anna Rogers, and Anna Rumshisky. 2019. Revealing the dark secrets of BERT. 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 4365–4374, Hong Kong, China, November. Association for Computational Linguistics.

Guillaume Lample, Alexandre Sablayrolles, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2019. Large memory layers with product keys. *arXiv preprint arXiv:1907.05242*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Stuart P. Lloyd. 1982. Least squares quantization in PCM. *IEEE Trans. Information Theory*, 28(2):129–136.

Paul Michel, Omer Levy, and Graham Neubig. 2019. Are sixteen heads really better than one? In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, pages 14014–14024.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, pages 3111–3119.

George A. Miller, Claudia Leacock, Randee Tengi, and Ross T. Bunker. 1993. A semantic concordance. In *HUMAN LANGUAGE TECHNOLOGY: Proceedings of a Workshop Held at Plainsboro, New Jersey, March 21-24, 1993*.

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

David Mimno and Laure Thompson. 2017. The strange geometry of skip-gram with negative sampling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2873–2878, Copenhagen, Denmark, September. Association for Computational Linguistics.

Jiaqi Mu and Pramod Viswanath. 2018. All-but-the-top: Simple and effective postprocessing for word representations. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*.

Roberto Navigli. 2006. Meaningful clustering of senses helps boost word sense disambiguation performance. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics*, ACL-44, pages 105–112, Stroudsburg, PA, USA. Association for Computational Linguistics.

Massimiliano Pavan and Marcello Pelillo. 2007. Dominant sets and pairwise clustering. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(1):167–172.

Marcello Pelillo. 2009. What is a cluster? perspectives from game theory. In *Proc. of the NIPS Workshop on Clustering Theory*.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, October. Association for Computational Linguistics.

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

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? 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 2463–2473, Hong Kong, China, November. Association for Computational Linguistics.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Alessandro Raganato, Jose Camacho-Collados, and Roberto Navigli. 2017. Word sense disambiguation: A unified evaluation framework and empirical comparison. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 99–110, Valencia, Spain, April. Association for Computational Linguistics.

Emily Reif, Ann Yuan, Martin Wattenberg, Fernanda B. Viégas, Andy Coenen, Adam Pearce, and Been Kim. 2019. Visualizing and measuring the geometry of BERT. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, pages 8592–8600.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601, Florence, Italy, July. Association for Computational Linguistics.

Rocco Tripodi and Roberto Navigli. 2019. Game theory meets embeddings: a unified framework for word sense disambiguation. 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 88–99, Hong Kong, China, November. Association for Computational Linguistics.

Rocco Tripodi and Stefano Li Pira. 2017. Analysis of italian word embeddings. In *Proceedings of the Fourth Italian Conference on Computational Linguistics (CLiC-it 2017), Rome, Italy, December 11-13, 2017*.

Rocco Tripodi, Sebastiano Vascon, and Marcello Pelillo. 2016. Context aware nonnegative matrix factorization clustering. In *23rd International Conference on Pattern Recognition, ICPR 2016, Cancún, Mexico, December 4-8, 2016*, pages 1719–1724.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium, November. Association for Computational Linguistics.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, pages 3261–3275.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *arXiv preprint arXiv:1906.08237*.