Sparsity Makes Sense: Word Sense Disambiguation Using Sparse Contextualized Word Representations

Gábor Berend[1,2]
[1]Institute of Informatics, University of Szeged
[2]MTA-SZTE Research Group on Aartificial Intelligence
berendg@inf.u-szeged.hu

Abstract

In this paper, we demonstrate that by utilizing sparse word representations, it becomes possible to surpass the results of more complex task-specific models on the task of fine-grained all-words word sense disambiguation. Our proposed algorithm relies on an overcomplete set of semantic basis vectors that allows us to obtain sparse contextualized word representations. We introduce such an information theory-inspired synset representation based on the co-occurrence of word senses and non-zero coordinates for word forms which allows us to achieve an aggregated F-score of 78.8 over a combination of five standard word sense disambiguating benchmark datasets. We also demonstrate the general applicability of our proposed framework by evaluating it towards part-of-speech tagging on four different treebanks. Our results indicate a significant improvement over the application of the dense word representations.

1 Introduction

Natural language processing applications have benefited remarkably form language modeling based contextualized word representations, including CoVe (McCann et al., 2017), ELMo (Peters et al., 2018) and BERT (Devlin et al., 2019), inter alia. Contrary to standard "static" word embeddings like `word2vec` (Mikolov et al., 2013) and Glove (Pennington et al., 2014), contextualized representations assign such vectorial representations to mentions of word forms that are sensitive to the entire sequence in which they are present. This characteristic of contextualized word embeddings makes them highly applicable for performing word sense disambiguation (WSD) as it has been investigated recently (Loureiro and Jorge, 2019; Vial et al., 2019).

Another popular line of research deals with sparse overcomplete word representations which differ from typical word embeddings in that most coefficients are exactly zero. Such sparse word representations have been argued to convey an increased interpretability (Murphy et al., 2012; Faruqui et al., 2015; Subramanian et al., 2018) which could be advantageous for WSD. It has been shown that sparsity can not only favor interpretability, but it can contribute to an increased performance in downstream applications (Faruqui et al., 2015; Berend, 2017).

The goal of this paper is to investigate and quantify what synergies exist between contextualized and sparse word representations. Our rigorous experiments show that it is possible to get increased performance on top of contextualized representations when they are post-processed in a way which ensures their sparsity.

In this paper we introduce an information theory-inspired algorithm for creating sparse contextualized word representations and evaluate it in a series of challenging WSD tasks. In our experiments, we managed to obtain solid results for multiple fine-grained word sense disambiguation benchmarks. All our source code for reproducing our experiments are made available at `https://github.com/begab/sparsity_makes_sense`.[1]

Our contributions can be summarized as follows:

- we propose the application of contextualized sparse overcomplete word representation in the task of word sense disambiguation,
- we carefully evaluate our information theory inspired approach for quantifying the strength of the connection between the individual dimensions of (sparse) word representations and

[1]An additional demo application performing all-words word sense disambiguation is also made available at `http://www.inf.u-szeged.hu/~berendg/nlp_demos/wsd`.

Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, pages 8498–8508,
November 16–20, 2020.

human interpretable semantic content such as fine grained word senses,

- we demonstrate the general applicability of our algorithm by applying it for POS tagging on four different UD treebanks.

2 Related work

One of the key difficulties of natural language understanding is the highly ambiguous nature of language. As a consequence, WSD has long-standing origins in the NLP community (Lesk, 1986; Resnik, 1997a,b), still receiving major recent research interest (Raganato et al., 2017a; Trask et al., 2015; Melamud et al., 2016; Loureiro and Jorge, 2019; Vial et al., 2019). A thorough survey on WSD algorithms of the pre-neural era can be found in (Navigli, 2009).

A typical evaluation for WSD systems is to quantify the extent to which they are capable of identifying the correct sense of ambiguous words in their contexts according to some sense inventory. One of the most frequently applied sense inventory in the case of English is the Princeton WordNet (Fellbaum, 1998) which also served the basis of our evaluation.

A variety of WSD approaches has evolved ranging from unsupervised and knowledge-based solutions to supervised ones. Unsupervised approaches could investigate the textual overlap between the context of ambiguous words and their potential sense definitions (Lesk, 1986) or they could be based on random walks over the semantic graph providing the sense inventory (Agirre and Soroa, 2009).

Supervised WSD techniques typically perform better than unsupervised approaches. IMS (Zhong and Ng, 2010) is a classical supervised WSD framework which was created with the intention of easy extensibility. It trains SVMs for predicting the correct sense of a word based on traditional features, such as surface forms and POS tags of the ambiguous words as well as its neighboring words.

The recent advent of neural text representations have also shaped the landscape of algorithms performing WSD. Iacobacci et al. (2016) extended the classical feature-based IMS framework by incorporating word embeddings. Melamud et al. (2016) devised context2vec, which relies on a bidirectional LSTM (biLSTM) for performing supervised WSD. Kågebäck and Salomonsson (2016) also proposed the utilization of biLSTMs for WSD. Raganato et al. (2017b) tackled all-words WSD as a sequence learning model and solved it using LSTMs. Vial et al. (2019) introduced a similar framework, but replaced the LSTM decoder with an ensemble of transformers. (Vial et al., 2019) additionally relied on BERT contextual word representations as input to their all-words WSD system.

Contextual word embeddings have recently superseded traditional word embeddings due to their advantageous property of also modeling the neighboring context of words upon determining their vectorial representations. As such, the same word form gets assigned a separate embedding when mentioned in different contexts. Contextualized word vectors, including (Devlin et al., 2019; Yang et al., 2019), typically employ some language modelling-inspired objective and are trained on massive amounts of textual data, which makes them generally applicable in a variety of settings as illustrated by top-performing entries at the SuperGLUE leaderboard (Wang et al., 2019).

Most recently, Loureiro and Jorge (2019) have proposed the usage of contextualized word representations for tackling WSD. Their framework builds upon BERT embeddings and performs WSD relying on a k-NN approach of query words towards the sense embeddings that are derived as the centroids of contextual embeddings labeled with a certain sense. The framework also utilizes static fasttext (Bojanowski et al., 2017) embeddings, and averaged contextual embeddings derived from the definitions attached to WordNet senses for mitigating the problem caused by the limited amounts of sense-labeled training data.

Kumar et al. (2019) proposed the EWISE approach which constructs sense definition embeddings also relying on the network structure of WordNet for performing zero-shot WSD in order to handle words without any sense-annotated occurrence in the training data. Bevilacqua and Navigli (2020) introduces EWISER as an improvement over the EWISE approach by providing a hybrid knowledge-based and supervised approach via the integration of explicit relational information from WordNet. Our approach differs from both (Kumar et al., 2019) and (Bevilacqua and Navigli, 2020) in that we are not exploiting the structural properties of WordNet.

SenseBERT (Levine et al., 2019) extends BERT (Devlin et al., 2019) by incorporating an auxiliary task into the masked language modeling objective for predicting word supersenses besides word iden-

tities. Our approach differs from SenseBERT as we do not propose an alternative way for training contextualized embeddings, but introduce an algorithm for extracting a useful representation from pretrained BERT embeddings that can effectively be used for WSD. Due to this conceptual difference, our approach does not need a large transformer model to be trained, but it can be steadily applied over pretrained models.

GlossBERT (Huang et al., 2019) framed WSD as a sentence pair classification task between the sentence containing an ambiguous target token and the contents of the glosses for the potential synsets of the ambiguous token and fine-tuned BERT accordingly. GlossBERT hence requires a fine-tuning stage, whereas our approach builds directly on the pre-trained contextual embeddings, which makes it more resource efficient.

Our work also relates to the line of research on sparse word representations. The seminal work on obtaining sparse word representations by Murphy et al. (2012) applied matrix factorization over the co-occurrence matrix built from some corpus. Arora et al. (2018) investigated the linear algebraic structure of static word embedding spaces and concluded that "simple sparse coding can recover vectors that approximately capture the senses". Faruqui et al. (2015); Berend (2017); Subramanian et al. (2018) introduced different approaches for obtaining sparse word representations from traditional static and dense word vectors. Our work differs from all the previously mentioned papers in that we create sparse *contextualized* word representations.

3 Approach

Our algorithm is composed of two important steps, i.e. we first make a sparse representation from the dense contextualized ones, then we derive a succinct representation describing the strength of connection between the individual basis of our representation and the sense inventory we would like to perform WSD against. We elaborate on these components next.

3.1 Sparse contextualized embeddings

Our algorithm first determines contextualized word representations for some sense-annotated corpus. We shall denote the surface form realizations in the corpus as $\mathcal{X} = \left\{ \left[x_j^{(i)} \right]_{j=0}^{N_i} \right\}_{i=0}^{M}$, with $x_j^{(i)}$ standing for the token at position j within sentence i, supposing a total of M sequences and N_i tokens in sentence i. We refer to the contextualized word representation for some token in boldface, i.e. $\boldsymbol{x}_j^{(i)}$ and the collection of contextual embeddings as $\mathbb{X} = \left\{ \left[\boldsymbol{x}_j^{(i)} \right]_{j=0}^{N_i} \right\}_{i=0}^{M}$.

Likewise to the sequence of sentences and their respective tokens, we also utilize a sequence of annotations that we denote as $\mathbb{S} = \left\{ \left[\boldsymbol{s}_j^{(i)} \right]_{j=0}^{N_i} \right\}_{i=0}^{M}$, with $\boldsymbol{s}_j^{(i)}$ indicating the labeling of token j within sentence i. We have $\boldsymbol{s}_j^{(i)} \in \{0,1\}^{|\mathcal{S}|}$ with $\mathcal{S}$ denoting the set of possible labels included in our annotated corpus. That is, we have an indicator vector conveying the annotation for every token. We allow for the $\boldsymbol{s}_j^{(i)} = \mathbf{0}$ case, meaning that it is possible that certain tokens lack annotation. In the case of WSD, the annotation is meant in the form of sense annotation, but in general, the token level annotations could convey other types of information as well.

The next step in our algorithm is to perform sparse coding over the contextual embeddings of the annotated corpus. Sparse coding is a matrix decomposition technique which tries to approximate some matrix $X \in \mathbb{R}^{v \times m}$ as a product of a sparse matrix $\alpha \in \mathbb{R}^{v \times k}$ and a dictionary matrix $D \in \mathbb{R}^{k \times m}$, where k denotes the number of basis vectors to be employed.

We formed matrix X by stacking and unit normalizing the contextual embeddings comprising $\mathbb{X}$. We then optimize

$$\min_{\substack{D \in \mathcal{C} \\ \boldsymbol{\alpha}_j^{(i)} \in \mathbb{R}^k_{\geq 0}}} \sum_{i=1}^{M} \sum_{j=1}^{N_i} \|\boldsymbol{x}_j^{(i)} - \boldsymbol{\alpha}_j^{(i)} D\|_2^2 + \lambda \|\boldsymbol{\alpha}_j^{(i)}\|_1, \quad (1)$$

where $\mathcal{C}$ denotes the convex set of matrices with row norm at most 1, λ is the regularization coefficient and the sparse coefficients in $\boldsymbol{\alpha}_j^{(i)}$ are required to be non-negative. We imposed the non-negativity constraint on α as it has been reported to provide increased interpretability (Murphy et al., 2012).

3.2 Binding basis vectors to senses

Once we have obtained a sparse contextualized representation for each token in our annotated corpus, we determine the extent to which the individual bases comprising the dictionary matrix D bind to the elements of our label inventory $\mathcal{S}$. In order to do so, we devise a matrix $\Phi \in \mathbb{R}^{k \times |\mathcal{S}|}$, which contains a ϕ_{bs} score for each pair of basis vector b and

a particular label s. We summarize our algorithm for obtaining Φ in Algorithm 1.

The definition of Φ is based on a generalization of co-occurrence of bases and the elements of the label inventory $\mathcal{S}$. We first define our co-occurrence matrix between bases and labels as

$$C = \sum_{i=1}^{M} \sum_{j=1}^{N_i} \boldsymbol{\alpha}_j^{(i)} \boldsymbol{s}_j^{(i)\intercal}, \tag{2}$$

i.e. C is the sum of outer products of sparse word representations ($\boldsymbol{\alpha}_j^{(i)}$) and their respective sense description vector ($\boldsymbol{s}_j^{(i)}$). The definition in (2) ensures that every $c_{bs} \in C$ aggregates the sparse nonnegative coefficients words labeled as s has received for their coordinate b. Recall that we allowed certain $\boldsymbol{s}_j^{(i)}$ to be the all zero vector, i.e. tokens that lack any annotation are conveniently handled by Eq. (2) as the sparse coefficients of such tokens do not contribute towards C.

We next turn the elements of C into a matrix representing a joint probability distribution P by determining the ℓ_1-normalized variant of C (line 5 of Algorithm 1). This way we devise a sparse matrix, the entries of which can be used for calculating Pointwise Mutual Information (PMI) between semantic bases and the presence of symbolic senses of our sense inventory.

For a pair of events (i, j) PMI is measured as $\log\left(\frac{p_{ij}}{p_{i*}p_{*j}}\right)$, with p_{ij} referring to their joint probability, p_{i*} and p_{*j} denoting the marginal probability of i and j, respectively. We determine these probabilities from the entries of P that we obtain from C via ℓ_1 normalization.

Employing Positive PMI Negative PMI values for a pair of events convey the information that they repel each other. Multiple studies have argued that negative PMI values are hence detrimental (Bullinaria and Levy, 2007; Levy et al., 2015) . To this end, we could opt for the determination of positive PMI (pPMI) values as indicated in line 7 of Algorithm 1.

Employing normalized PMI An additional property of (positive) PMI is that it favors observations with low marginal frequency (Bouma, 2009), since for events with low $p(x)$ marginal probability $p(x|y) \approx p(x)$ tend to hold, which results in high PMI values. In our setting, it would result in rarer senses receiving higher ϕ_{bs} scores towards all the bases.

In order to handle low-frequency senses better, we optionally calculate the normalized (positive) PMI (Bouma, 2009) between a pair of base and sense as $\log\left(\frac{p_{ij}}{p_{i*}p_{*j}}\right) \Big/ -\log(p_{ij})$. That is, we normalize the PMI scores by the negative logarithm of the joint probability (cf. line 8 of Algorithm 1). This step additionally ensures that the normalized PMI (nPMI) ranges between -1 and 1 as opposed to the $(-\infty, \min(-\log(p_i), -\log(p_j)))$ range of the unnormalized PMI values.

Algorithm 1 Calculating Φ

Require: sense annotated corpus $(\mathbb{X}, \mathbb{S})$
Ensure: $\Phi \in \mathbb{R}^{k \times |S|}$ describing the strength between k sense basis and the elements of the sense inventory $|S|$

1: **procedure** CALCULATEPHI($\mathbb{X}, \mathbb{S}$)
2: $X \leftarrow$ UNITNORMALIZE($\mathbb{X}$)
3: $D, \alpha \leftarrow \underset{D \in \mathcal{C}, \alpha \in \mathbb{R}_{\geq 0}}{\arg\min} \|X - D\alpha\|_F + \lambda\|\alpha\|_1$
4: $C \leftarrow \alpha S$
5: $P \leftarrow C / \|C|_1$
6: $\Phi \leftarrow \left[\log\left(\frac{p_{ij}}{p_{i*}p_{*j}}\right)\right]_{ij}$
7: $\Phi \leftarrow \left[\max(0, \phi_{ij})\right]_{ij}$ ▷ cf. pPMI
8: $\Phi \leftarrow \left[\frac{\phi_{ij}}{-\log(p_{ij})}\right]_{ij}$ ▷ cf. nPMI
9: **return** Φ, D
10: **end procedure**

3.3 Inferring senses

We now describe the way we assign the most plausible sense to any given token from a sequence according to the sense inventory employed for constructing D and Φ.

For an input sequence of N tokens accompanied by their corresponding contextualized word representations as $[\boldsymbol{x}_j]_{j=1}^N$, we determine their corresponding sparse representations $[\boldsymbol{\alpha}_j]_{j=1}^N$ based on D that we have already determined upon obtaining Φ. That is, we solve an ℓ_1-regularized convex optimization problem with D being kept fixed for all the unit normalized vectors $\boldsymbol{x}_j$ in order to obtain the sparse contextualized word representation $\boldsymbol{\alpha}_j$ for every token j in the sequence.

We then take the product between $\boldsymbol{\alpha}_j \in \mathbb{R}^k$ and $\Phi \in \mathbb{R}^{k \times |\mathcal{S}|}$. Since every column in Φ corresponds to a sense from the sense inventory, every scalar in the resulting product $\boldsymbol{\alpha}_j^\intercal \Phi \in \mathbb{R}^{|\mathcal{S}|}$ can be interpreted as the quantity indicating the extent to which token j – in its given context – pertains to the in-

dividual senses from the sense inventory. In other words, we assign that sense s to a particular token j which maximizes $\boldsymbol{\alpha}_j^\intercal \Phi_{*s}$, where Φ_{*s} indicates the column vector from Φ corresponding to sense s.

4 Experiments and results

We evaluate our approach towards the unified WSD evaluation framework released by Raganato et al. (2017a) which includes the sense-annotated SemCor dataset for training purposes. SemCor (Miller et al., 1994) consists of 802,443 tokens with more than 28% (226,036) of its tokens being sense-annotated using WordNet sensekeys.

For instance `bank%1:14:00::` is one of the possible sensekeys the word *bank* can be assigned to according to one of the 18 different synsets it is included in WordNet 3.0. WordNet 3.0 contains all together 206,949 distinct senses for 147,306 unique lemmas grouped into 117,659 synsets. We constructed Φ relying on the synset-level information of WordNet.

4.1 Sparse contextualized embeddings

For obtaining contextualized word representations, we rely on the pretrained `bert-large-cased` model from (Wolf et al., 2019). Each input token $x_j^{(i)}$ gets assigned 25 contextual vectors $[\boldsymbol{x}_{j,l}^{(i)}]_{l=0}^{24}$ according to the input and the 24 inner layers of the BERT-large model. Each vector $\boldsymbol{x}_{j,l}^{(i)}$ is 1024-dimensional.

BERT relies on WordPiece tokenization, which means that a single token, such as *playing*, could be broken up into multiple subwords (*play* and *##ing*). We defined token-level contextual embeddings to be the average of their subword-level contextual embeddings.

Sparse coding as formulated in (1) took the stacked 1024-dimensional contextualized BERT embeddings for the 802,443 tokens from SemCor as input, i.e. we had $X \in \mathbb{R}^{1024 \times 802443}$. We used the SPAMS library (Mairal et al., 2009) to solve our optimization problems. Our approach has two hyperparameters, i.e. the number of basis vectors included in the dictionary matrix (k) and the regularization coefficient (λ). We experimented with $k \in \{1500, 2000, 3000\}$ in order to investigate the sensitivity of our proposed algorithm towards the dimension of the sparse vectors and we employed $\lambda = 0.05$ throughout all our experiments.

Figure 1 includes the average number of nonzero coefficients for the sparse word representations from the SemCor database when using different values of k and different layers of BERT as input. The average time for determining sparse contextual word representations for one layer of BERT was 40 minutes on an Intel Xeon 5218 for $k = 3000$.

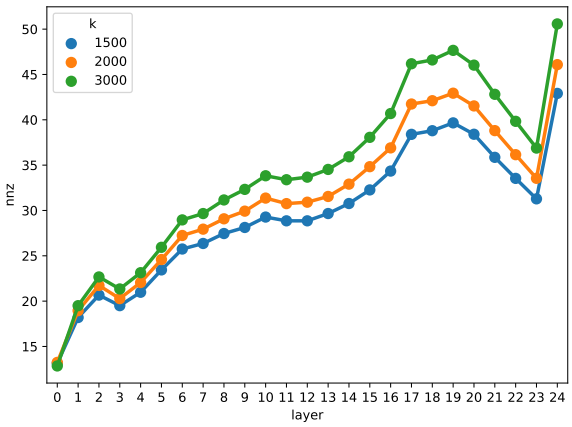

Figure 1: Average number of nonzero coefficients per SemCor tokens when relying on contextualized embeddings from different layers of BERT as input.

4.2 Evaluation on all-words WSD

The evaluation framework introduced in (Raganato et al., 2017a) contains five different all-words WSD benchmarks for measuring the performance of WSD systems. The dataset includes the SensEval2 (Edmonds and Cotton, 2001), SensEval3 (Mihalcea et al., 2004), SemEval 2007 Task 17 (Pradhan et al., 2007), SemEval 2013 Task 12 (Navigli et al., 2013), SemEval 2015 Task 13 (Moro and Navigli, 2015) datasets each containing 2282, 1850, 455, 1644 and 1022 sense annotated tokens, respectively.

The concatenation of the previous datasets is also included in the evaluation toolkit, which is commonly referred as the ALL dataset that includes 7253 sense-annotated test cases. We relied on the official scoring script included in the evaluation framework from (Raganato et al., 2017a). Unless stated otherwise, we report our results on the combination of all the datasets for brevity as results for all the subcorpora behaved similarly.

In order to demonstrate the benefits of our proposed approach, we develop a strong baseline similar to the one devised in (Loureiro and Jorge, 2019). This approach employs the very same contextualized embeddings that we use otherwise in our algorithm for providing identical conditions for the different approaches. For each synset s, we then determine its centroid based on the contextualized word representations pertaining to sense s accord-

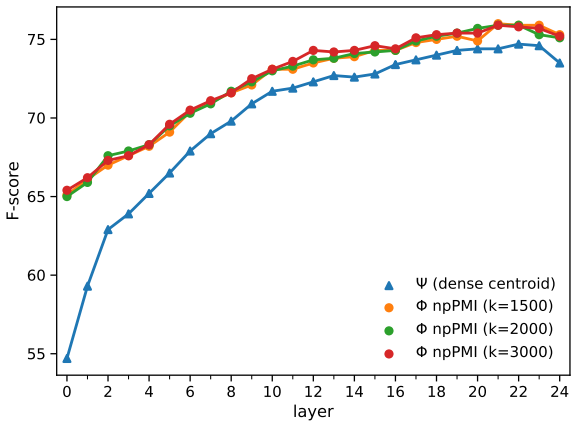

Figure 2: Comparative results of relying on the dense and sparse word representations of different dimensions for WSD using the SemCor dataset for training.

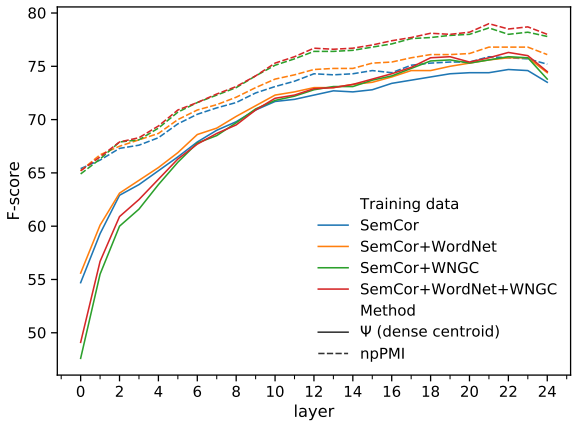

Figure 3: The effects of employing additional sources of information besides SemCor during training.

ing to the training data. We then use this matrix Ψ as a replacement over Φ when making predictions for some token with its dense contextualized embedding $\boldsymbol{x}_j$.

The way we make our fine-grained sensekey predictions towards the test tokens are identical when utilizing dense and sparse contextualized embeddings, the only difference is whether we base our decision on $\boldsymbol{x}_j^\intercal \Psi$ (for the dense case) or $\boldsymbol{\alpha}_j^\intercal \Phi$ (for the sparse case). In either case, we choose the best scoring synset a particular query lemma can belong to. That is, we perform argmax operation described in Section 3.3 over the set of possible synsets a query lemma can belong to.

Figure 2 includes comparative results for the approach using dense and sparse contextualized embeddings derived from different layers of BERT. We can see that our approach yields considerable improvements over the application of dense embeddings. In fact, applying sparse contextualized embeddings provided significantly better results ($p \ll 0.01$ using McNemar's test) irrespective of the choice of k when compared against the utilization of dense embeddings.

Additionally, the different choices for the dimension of the sparse word representations does not seem to play a decisive role as illustrated by Figure 2 and also confirmed by our significance tests conducted between the sparse approaches using different values of k. Since the choice of k does not severely impacted results, we report our experiments for the $k = 3000$ case hereon.

4.2.1 Increasing the amount of training data

We also measured the effects of increasing the amount of training data. We additionally used two sources of information, i.e. the WordNet synsets themselves and the Princeton WordNet Gloss Corpus (WNGC) for training. The WordNet synsets were utilized in an identical fashion to the LMMS approach (Loureiro and Jorge, 2019), i.e. we determined a vectorial representation for each synset by taking the average of the contextual representations that based on the concatenation of the definition and the lemmas belonging to the synsets.

WNGC includes a sense-annotated version of WordNet itself containing 117,659 definitions (one for each synset in WordNet), consisting of 1,634,691 tokens out of which 614,435 has a corresponding sensekey attached to. We obtained this data from the Unification of Sense Annotated Corpora (UFSAC) (Vial et al., 2018).

For this experiment all our framework was kept intact, the only difference was that instead of solely relying on the sense-annotated training data included in SemCor, we additionally relied on the sense representations derived from WordNet glosses and sense annotations included in WNGC upon the determination of Φ and Ψ for the sparse and dense cases, respectively. For these experiments we used the same set of semantic basis vectors D that we determined earlier for the case when we relied solely on SemCor as the source of sense annotated dataset. Figure 3 includes our results when increasing the amount of sense-annotated training data. We can see that the additional training data consistently improves performance for both the dense and the sparse case. Figure 3 demon-

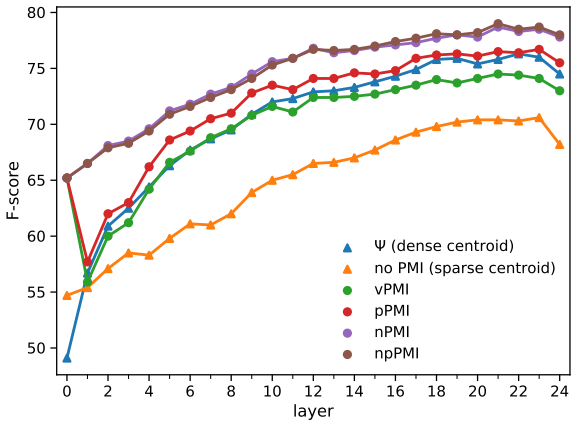

Figure 4: Ablation experiments regarding the different strategies to calculate Φ using the combined (SemCor+WordNet+WNGC) training data.

strates that our proposed method when trained on the SemCor data alone is capable of achieving the same or better performance as the approach which is based on dense contextual embeddings using all the available sources of training signal.

4.2.2 Ablation experiments

We gave a detailed description of our algorithm in Section 3.2. We now report our experimental results that we conducted in order to see the contribution of the individual components of our algorithms. As mentioned in Section 3.2, determining normalized positive PMI (`npPMI`) between the semantic bases and the elements of the sense inventory plays a central role in our algorithm.

In order to see the effects of normalizing and keeping only the positive PMI values, we evaluated 3 further *PMI-based variants for the calculation of Φ, i.e. we had

- `vPMI` vanilla PMI without normalization or discarding negative entries,
- `pPMI`, which discards negative PMI values but does not normalize them and
- `nPMI` which performs normalization, however does not discard negative PMI values.

Additionally, we evaluated the system which uses sparse contextualized word representations for determining Φ, however, does not involve the calculation of PMI scores at all. In that case we calculated a centroid for every synset similar to the calculation of Ψ for the case of contextualized embeddings that are kept dense. The only difference is that for the approach we refer to as `no PMI`, we calculated synset centroids based on the sparse contextualized word representations.

Figure 4 includes our results for the previously mentioned variants of our algorithm when relying on the different layers of BERT as input. Figure 4 highlights that calculating PMI is indeed a crucial step in our algorithm (cf. the `no PMI` and `*PMI` results). We also tried to adapt the `*PMI` approaches for the dense contextual embeddings, but the results dropped severely in that case.

We can additionally observe that normalization has the most impact on improving the results, as the performance of `nPMI` is at least 4 points better than that of `vPMI` for all layers. Not relying on negative PMI scores also had an overall positive effect (cf. `vPMI` and `pPMI`), which seems to be additive with normalization (cf. `nPMI` and `npPMI`).

4.2.3 Comparative results

We next provide detailed performance results broken down for the individual subcorpora of the evaluation dataset. Table 1 includes comparative results to previous methods that also use SemCor and optionally WordNet glosses as their training data. In Table 1 we report our results obtained by our model which derives sparse contextual word embeddings based on the averaged representations retrieved from the last four layers of BERT identical to how it was done in (Loureiro and Jorge, 2019). Figure 4 illustrates that reporting results from any of the last 4 layers would not change our overall results substantially.

Table 1 reveals that it is only the LMMS_{2348} (Loureiro and Jorge, 2019) approach which performs comparably to our algorithm. LMMS_{2348} determines dense sense representations relying on the large BERT model as well. The sense representations used by LMMS_{2348} are a concatenation of the 1024-dimensional centroids of each senses encountered in the training data, an 1024-dimensional vectors derived from the glosses of WordNet synsets and a 300-dimensional static fasttext embeddings. Even though our approach does not rely on static fasttext embeddings, we still managed to improve upon the best results reported in (Loureiro and Jorge, 2019). The improvement of our approach which uses the SemCor training data alone is 1.9 points compared to the LMMS_{1024}, i.e. such a variant of the LMMS system (Loureiro and Jorge, 2019) which also relies solely on BERT representations for the SemCor training set.

approach	SensEval2	SensEval3	SemEval2007	SemEval2013	SemEval2015	ALL
Most Frequent Sense (MFS)	66.8	66.2	55.2	63.0	67.8	65.2
IMS (Zhong and Ng, 2010)	70.9	69.3	61.3	65.3	69.5	68.4
IMS+emb-s (Iacobacci et al., 2016)	72.2	70.4	62.6	65.9	71.5	69.6
context2Vec (Melamud et al., 2016)	71.8	69.1	61.3	65.6	71.9	69.0
LMMS_{1024} (Loureiro and Jorge, 2019)	75.4	74.0	66.4	72.7	75.3	73.8
LMMS_{2348} (Loureiro and Jorge, 2019)	76.3	75.6	68.1	75.1	77.0	75.4
GlossBERT(Sent-CLS-WS) (Huang et al., 2019)	77.7	75.2	72.5	76.1	80.4	77.0
Ours (using SemCor)	77.6	76.8	68.4	73.4	76.5	75.7
Ours (using SemCor + WordNet)	77.9	**77.8**	68.8	76.1	77.5	76.8
Ours (using SemCor + WordNet + WNGC)	**79.6**	77.3	**73.0**	**79.4**	**81.3**	**78.8**

Table 1: Comparison with previous supervised results in terms of F measure computed by the official scorer provided in (Raganato et al., 2017a).

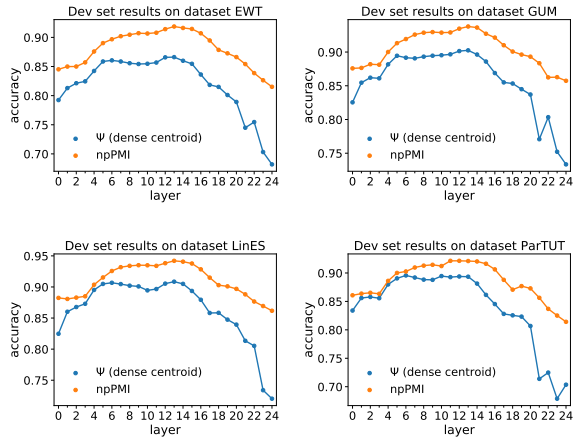

Figure 5: POS tagging results evaluated over the development set of four English UD v2.5 treebanks.

Treebank	Centroid (Ψ)	npPMI (Φ)	p-value
EWT	86.66	**91.81**	7e-193
GUM	89.58	**92.93**	2e-63
LinES	91.24	**94.64**	1e-87
ParTUT	90.73	**92.99**	4e-7

Table 2: Comparison of the adaptation of the LMMS approach and ours on POS tagging over the test sets of four English UD v2.5 treebanks. The last column contains the p-value for the McNemar test comparing the different behavior of the two approaches.

4.3 Evaluation towards POS tagging

In order to demonstrate the general applicability of our proposed algorithm, we evaluated it towards POS tagging using version 2.5 of Universal Dependencies. We conducted experiments over four different subcorpora in English, namely the EWT (Silveira et al., 2014), GUM (Zeldes, 2017), LinEs (Ahrenberg, 2007) and ParTut (Sanguinetti and Bosco, 2015) treebanks.

For these experiments, we used the same approach as before. We also used the same dictionary matrix D for obtaining the sparse word representations that we determined based on the SemCor dataset. The only difference for our POS tagging experiments is that this time the token level labels were replaced by the POS tags of the individual tokens as opposed to their sense labels. This means that both Ψ and Φ had 17 columns, i.e. the number of distinct POS tags used in these treebanks.

Figure 5 reveals that the approach utilizing sparse contextualized word representations outperform the one that is based on the adaptation of the LMMS approach for POS tagging by a fair margin, again irrespective of the layer of BERT that is used as input. A notable difference compared to the results obtained for all-words WSD that for POS tagging the intermediate layers of BERT seem to deliver the most useful representation.

We used the development set of the individual treebanks for choosing the most promising layer of BERT to employ the different approaches over. For the `npPMI` approach we selected layer 13, 13, 14 and 11 for the EWT, GUM, LinES and ParTut treebanks. As for the dense centroid based approach we selected layer 6 for the ParTUT treebank and layer 13 for the rest of the treebanks. After doing so, our results for the test set of the four treebanks are reported in Table 2. Our approach delivered significant improvements for POS tagging as well as indicated by the p-values of the McNemar test.

5 Conclusions

In this paper we investigated how the application of sparse word representations obtained from contextualized word embeddings can provide a substantially increased ability for solving problems that require the distinction of fine-grained word

senses. In our experiments, we managed to obtain solid results for multiple fine-grained word sense disambiguation benchmarks with the help of our information theory-inspired algorithm. We additionally carefully investigated the effects of increasing the amount of sense-annotated training data and the different design choices we made. We also demonstrated the general applicability of our approach by evaluating it in POS tagging. Our source code is made available at https://github.com/begab/sparsity_makes_sense.

Acknowledgments

This work was in part supported by the National Research, Development and Innovation Office of Hungary through the Artificial Intelligence National Excellence Program (grant no.: 2018-1.2.1-NKP-2018-00008).

References

Eneko Agirre and Aitor Soroa. 2009. Personalizing PageRank for word sense disambiguation. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, EACL '09, pages 33–41, Stroudsburg, PA, USA. Association for Computational Linguistics.

Lars Ahrenberg. 2007. LinES: An English-Swedish parallel treebank. In *Proceedings of the 16th Nordic Conference of Computational Linguistics (NODALIDA 2007)*, pages 270–273, Tartu, Estonia. University of Tartu, Estonia.

Sanjeev Arora, Yuanzhi Li, Yingyu Liang, Tengyu Ma, and Andrej Risteski. 2018. Linear algebraic structure of word senses, with applications to polysemy. *Transactions of the Association for Computational Linguistics*, 6:483–495.

Gábor Berend. 2017. Sparse coding of neural word embeddings for multilingual sequence labeling. *Transactions of the Association for Computational Linguistics*, 5:247–261.

Michele Bevilacqua and Roberto Navigli. 2020. Breaking through the 80in word sense disambiguation by incorporating knowledge graph information. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

G. Bouma. 2009. Normalized (pointwise) mutual information in collocation extraction. In *From Form to Meaning: Processing Texts Automatically, Proceedings of the Biennial GSCL Conference 2009*, volume Normalized, pages 31–40, Tübingen.

John A. Bullinaria and Joseph P. Levy. 2007. Extracting semantic representations from word co-occurrence statistics: A computational study. *Behavior Research Methods*, 39(3):510–526.

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.

Philip Edmonds and Scott Cotton. 2001. SENSEVAL-2: Overview. In *The Proceedings of the Second International Workshop on Evaluating Word Sense Disambiguation Systems*, SENSEVAL '01, pages 1–5, Stroudsburg, PA, USA. Association for Computational Linguistics.

Manaal Faruqui, Yulia Tsvetkov, Dani Yogatama, Chris Dyer, and Noah A. Smith. 2015. Sparse overcomplete word vector representations. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1491–1500, Beijing, China. Association for Computational Linguistics.

Christiane Fellbaum. 1998. *WordNet: An Electronic Lexical Database*. Bradford Books.

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.

Ignacio Iacobacci, Mohammad Taher Pilehvar, and Roberto Navigli. 2016. Embeddings for word sense disambiguation: An evaluation study. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 897–907, Berlin, Germany. Association for Computational Linguistics.

Mikael Kågebäck and Hans Salomonsson. 2016. Word sense disambiguation using a bidirectional LSTM. In *Proceedings of the 5th Workshop on Cognitive Aspects of the Lexicon (CogALex - V)*, pages 51–56, Osaka, Japan. The COLING 2016 Organizing Committee.

Sawan Kumar, Sharmistha Jat, Karan Saxena, and Partha Talukdar. 2019. Zero-shot word sense disambiguation using sense definition embeddings. In

Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 5670–5681, Florence, Italy. Association for Computational Linguistics.

Michael Lesk. 1986. Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone. In *Proceedings of the 5th Annual International Conference on Systems Documentation*, SIGDOC '86, pages 24–26, New York, NY, USA. ACM.

Yoav Levine, Barak Lenz, Or Dagan, Dan Padnos, Or Sharir, Shai Shalev-Shwartz, Amnon Shashua, and Yoav Shoham. 2019. SenseBERT: Driving some sense into BERT. *CoRR*, abs/1908.05646.

Omer Levy, Yoav Goldberg, and Ido Dagan. 2015. Improving distributional similarity with lessons learned from word embeddings. *Transactions of the Association for Computational Linguistics*, 3:211–225.

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.

Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro. 2009. Online dictionary learning for sparse coding. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 689–696, New York, NY, USA. ACM.

Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. 2017. Learned in translation: Contextualized word vectors. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 6294–6305. Curran Associates, Inc.

Oren Melamud, Jacob Goldberger, and Ido Dagan. 2016. context2vec: Learning generic context embedding with bidirectional LSTM. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 51–61, Berlin, Germany. Association for Computational Linguistics.

Rada Mihalcea, Timothy Chklovski, and Adam Kilgarriff. 2004. The senseval-3 english lexical sample task. In *Proceedings of SENSEVAL-3, the Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, pages 25–28, Barcelona, Spain. Association for Computational Linguistics.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc.

George A. Miller, Martin Chodorow, Shari Landes, Claudia Leacock, and Robert G. Thomas. 1994. Using a semantic concordance for sense identification. In *HUMAN LANGUAGE TECHNOLOGY: Proceedings of a Workshop held at Plainsboro, New Jersey, March 8-11, 1994*.

Andrea Moro and Roberto Navigli. 2015. SemEval-2015 task 13: Multilingual all-words sense disambiguation and entity linking. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 288–297, Denver, Colorado. Association for Computational Linguistics.

Brian Murphy, Partha Talukdar, and Tom Mitchell. 2012. Learning effective and interpretable semantic models using non-negative sparse embedding. In *Proceedings of COLING 2012*, pages 1933–1950, Mumbai, India. The COLING 2012 Organizing Committee.

Roberto Navigli. 2009. Word sense disambiguation: A survey. *ACM Comput. Surv.*, 41(2):10:1–10:69.

Roberto Navigli, David Jurgens, and Daniele Vannella. 2013. SemEval-2013 task 12: Multilingual word sense disambiguation. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 222–231, Atlanta, Georgia, USA. Association for Computational Linguistics.

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. 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. Association for Computational Linguistics.

Sameer S. Pradhan, Edward Loper, Dmitriy Dligach, and Martha Palmer. 2007. Semeval-2007 task 17: English lexical sample, srl and all words. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, SemEval '07, pages 87–92, Stroudsburg, PA, USA. Association for Computational Linguistics.

Alessandro Raganato, Jose Camacho-Collados, and Roberto Navigli. 2017a. 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. Association for Computational Linguistics.

Alessandro Raganato, Claudio Delli Bovi, and Roberto Navigli. 2017b. Neural sequence learning models for word sense disambiguation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1156–1167, Copenhagen, Denmark. Association for Computational Linguistics.

Philip Resnik. 1997a. A perspective on word sense disambiguation methods and their evaluation. In *Tagging Text with Lexical Semantics: Why, What, and How?*

Philip Resnik. 1997b. Selectional preference and sense disambiguation. In *Tagging Text with Lexical Semantics: Why, What, and How?*

Manuela Sanguinetti and Cristina Bosco. 2015. Parttut: The turin university parallel treebank. In *Harmonization and Development of Resources and Tools for Italian Natural Language Processing within the PARLI Project*, pages 51–69.

Natalia Silveira, Timothy Dozat, Marie-Catherine de Marneffe, Samuel Bowman, Miriam Connor, John Bauer, and Christopher D. Manning. 2014. A gold standard dependency corpus for English. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014)*.

Anant Subramanian, Danish Pruthi, Harsh Jhamtani, Taylor Berg-Kirkpatrick, and Eduard H. Hovy. 2018. SPINE: sparse interpretable neural embeddings. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 4921–4928.

Andrew Trask, Phil Michalak, and John Liu. 2015. sense2vec - A fast and accurate method for word sense disambiguation in neural word embeddings. *CoRR*, abs/1511.06388.

Loïc Vial, Benjamin Lecouteux, and Didier Schwab. 2018. UFSAC: Unification of sense annotated corpora and tools. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018)*, Miyazaki, Japan. European Languages Resources Association (ELRA).

Loïc Vial, Benjamin Lecouteux, and Didier Schwab. 2019. Sense Vocabulary Compression through the Semantic Knowledge of WordNet for Neural Word Sense Disambiguation. In *Global Wordnet Conference*, Wroclaw, Poland.

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.

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

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. Cite arxiv:1906.08237Comment: Pretrained models and code are available at https://github.com/zihangdai/xlnet.

Amir Zeldes. 2017. The GUM corpus: Creating multilayer resources in the classroom. *Language Resources and Evaluation*, 51(3):581–612.

Zhi Zhong and Hwee Tou Ng. 2010. It makes sense: A wide-coverage word sense disambiguation system for free text. In *Proceedings of the ACL 2010 System Demonstrations*, pages 78–83, Uppsala, Sweden. Association for Computational Linguistics.