Abstract

Word sense induction (WSI) is a difficult problem in natural language processing that involves the unsupervised automatic detection of a word’s senses (i.e. meanings). Recent work achieves significant results on the WSI task by pre-training a language model that can exclusively disambiguate word senses, whereas others employ previously pre-trained language models in conjunction with additional strategies to induce senses. In this paper, we propose a novel unsupervised method based on hierarchical clustering and invariant information clustering (IIC). The IIC is used to train a small model to optimize the mutual information between two vector representations of a target word occurring in a pair of synthetic paraphrases. This model is later used in inference mode to extract a higher quality vector representation to be used in the hierarchical clustering. We evaluate our method on two WSI tasks and in two distinct clustering configurations (fixed and dynamic number of clusters). We empirically demonstrate that, in certain cases, our approach outperforms prior WSI state-of-the-art methods, while in others, it achieves a competitive performance.

1 Introduction

The automatic identification of a word’s senses is an open problem in natural language processing, known as "word sense induction" (WSI). It is clear that the task of word sense induction is closely related to the task of word sense disambiguation (WSD) relying on a predefined sense inventory (i.e. WordNet (Fellbaum, 1998; Wallace, 2007; Feinerer and Hornik, 2020)) and aiming to solve the word’s ambiguity in context. In WSI, given a target word, we focus on clustering a collection of sentences using this word according to its senses. For example, figure 1 shows the different clusters obtained by using RoBERTa_{LARGE} (Liu et al., 2019) of 3000 sentences that contain the word bank collected from Wikipedia. We can see five different clusters where the corresponding centroids represent the 2D PCA projection of the average contextual word vectors of the word bank. The clusters are obtained using the agglomerative clustering with cosine affinity and average linkage. Word senses are more beneficial than simple word forms for a variety of tasks including Information Retrieval, Machine Translation and others (Pantel and Lin, 2002). The former are typically represented as a fixed list of definitions from a manually constructed lexical database. However, lexical databases are missing important domain-specific senses. For example, these databases often lack explicit semantic or contextual links between concepts and definitions (Agirre et al., 2009). Hand-crafted lexical databases also frequently fail to convey the precise meaning of a target word in a specific context (Véronis, 2004). In order to address these issues, WSI intends to learn in an unsupervised manner the various meanings of a given word.

This paper includes the following contributions:

1) We propose a new unsupervised method using contextual word embeddings (i.e. RoBERTa, BERT and DeBERTa (He et al., 2021)) that are being updated with more sense-related information by maximizing the mutual information between two instances of the same cluster. This method presents competitive results on WSI tasks.

2) We apply for the first time a method to compute a dynamic number of senses for each word. We rely on a recent word polysemy score function (Xypolopoulos et al., 2020).

3) We study the sense information per hidden layer for four different pretrained language models. We share, for all models, the layers with the best performance on sense-related tasks.
2 Related Work

Previous works on WSI use generative statistical models to solve this task. Mainly, they approach this task as a topic modeling problem using Latent Dirichlet Allocation (LDA) (Lau et al., 2012; Chang et al., 2014; Goyal and Hovy, 2014; Wang et al., 2015; Komninos and Manandhar, 2016). AutoSense (Amplayo et al., 2018), one of the most recent best-performing LDA methods, is based on two principles: First, senses are represented as a distribution over topics. Second, the model generates a pair composed of the target word and its neighboring word, thus separating the topic distributions into fine-grained senses based on lexical semantics. AutoSense throws away the garbage senses by removing topics distributions that don’t belong to any instance. Furthermore, it adds new ones according to the generated (target, neighbor) pairs which means that fixing the number of senses by the model is not required. While most of the WSI methods fix the number of clusters for all the words, in our work we explore two setups for the number of clusters, fixed and dynamic. Other works (Song et al., 2016; Corrêa and Amancio, 2018) use the static word embedding Word2Vec (Mikolov et al., 2013) to get the representations of polysemous words before applying the clustering method.

After the emergence of contextual word Embeddings, pretrained language models such as ELMo (Peters et al., 2018) (based on BiLSTM) and BERT (Devlin et al., 2019) (based on the transformers) (Vaswani et al., 2017) are used with additional techniques to induce senses of a target word. (Amrami and Goldberg, 2018) and (Amrami and Goldberg, 2019) use consecutively ELMo and BERT-LARGE to predict probable substitutes for the target words. Next, it gives each instance $k$ representatives where each one contains multiple possible substitutes drawn randomly from the word distribution predicted by the language model. Each representative is a vector conducted from TF-IDF. Following, the representatives are clustered using the agglomerative clustering where the number of clusters is fixed to 7. Finally, each instance will be assigned to one or multiple clusters according to the corresponding cluster of each of its representatives. Instead of using the word substitutes approach, our work uses the contextual word embedding extracted from pre-
PolyLM (Ansell et al., 2021) is one of the most recent techniques for word sense induction that uses a MLM (Masked Language Model) to induce senses. PolyLM took a novel approach to the problem of learning word senses. It uses the transformer architecture to predict eight probabilities for each word, where each probability represents the probability of a word to be assigned to one of eight different senses. It is built on two assumptions: the chance of a word being predicted in a masked place is proportional to the total of its distinct senses, and for a particular context, one of the word’s senses is more likely to be used. The model has the drawback of assuming the same fixed number of senses for all words.

3 Method

Our method consists of four main steps: First, we form a dataset of pair of sentences. To guarantee the fact that the target word in one pair belongs to the same cluster-based sense, each sentence is joined with a randomly perturbed replicate. Further details can be found in 3.1. Second, we extract the pairwise hidden state representations of the target word from one of the layers of a pretrained language model. Hence, for each pair of sentences, we have now a pair of vectors. Mainly, we will evaluate this method on RoBERTa$^L$AR$^L$GE, BERT$^L$AR$^L$GE and DeBERTa$^m$L$^L$AR$^L$GE. Third, we train a MIM (Mutual Information Maximization) model where: (1) Considering an instance of the pairwise hidden state representations, the network is trained with an objective function maximizing the mutual information and minimizing the match loss between the output of the two vectors. (2) The best instance of the model is chosen according to the least loss on the pre-defined test set. (3) The original vectors of the test set are now replaced with the output of the first layer as the new vectors for the target word. Fourth, for each target word in the evaluation datasets, we apply the agglomerative clustering method on the new outputs to obtain our clustering solution. To choose the pre-defined number of clusters, we follow two approaches: (i) Fix the number of senses (clusters) to 7 as in Amrami and Goldberg (2018, 2019) and (ii) Use a dynamic number of clusters based on the polysemy score (Xypolopoulos et al., 2020) of each target word. The main steps are detailed in the following subsections.

Figure 2: The pipeline of our method: For the word “live” chosen as target, a list of sentences is provided. BART is used to generate their corresponding paraphrases. The hidden representation $X_{live}$ of the target word is extracted from the layer $l$ of a pretrained language model. Dashed line denotes shared parameters.

3.1 Dataset Setup

BART (Lewis et al., 2020) is a denoising autoencoder for pretraining sequence-to-sequence models. It is trained by learning a model to rebuild a corrupted version of the original sentences using an arbitrary noising function. It is based on a standard Transformer-based neural machine translation architecture which can be seen as a generalization of BERT (due to the bidirectional encoder), GPT (Radford and Narasimhan, 2018) (with the left-to-right decoder), and other recent pretraining schemes. BART can be used also as a generative model given an input i.e. sentence completion, translation, summarization, etc..

Generating randomly perturbated replicates

In order to apply our method on the text input, we need to create a pair of sentences where the target word has the same sense. To fulfil this, a function is needed to introduce random perturbations to the input sentence while preserving the meaning. The sentence and its perturbated version are keeping the same sense of the target lemma. Thus, we can generate a pair of sentences that belong to the same cluster. First, we masked 40% of the original sentence while preventing -in most cases- masking the target word. Second, we predicted the masked tokens using BART$^B$BASE with a beam size of one.
3.2 Vectors Extraction

The train set is used to train the parameters of a small network while the test set is used to perform the induction of the senses. Using the best layer of each of the following transformer-based models: BERT\textsubscript{LARGE}, RoBERTa\textsubscript{LARGE} and DeBERTa\textsubscript{XLARGE}, we extracted representations of the different train and test instances. The best layer for each pretrained language model is chosen according to the best performance on BERTScore (Zhang* et al., 2020) with WMT16 To-English Pearson\textsuperscript{1}.

At this stage, if the target word is broken down into pieces. Note that, while generating the perturbation on the input text using BART\textsubscript{BASE}, there is a small probability that the paraphrase might not contain the target word. Thus, all the sentences in the training set with their corresponding paraphrases deprived of the target word are removed.

3.3 Loss Function

We seek to minimize a loss function $L$ with two components, each of which is explained in the following:

$$L = L_{IIC} + L_M$$

3.3.1 Invariant Information Clustering Loss

Invariant information clustering IIC (Ji et al., 2019) is a clustering objective that learns a neural network from scratch to perform unsupervised image classification and segmentation. The model learns to cluster unlabeled data based on maximizing the mutual information score between the unlabeled sample and a transformation of the input. Therefore, both the input and its corresponding transformation surely contain the same information and do belong to the same class/cluster. Maximizing the mutual information is robust to clustering degeneracy where a single cluster tends to dominate the predictions or some clusters tend to disappear as in k-means. Also, it helps to avoid noisy data from affecting the predictions by over-clustering. The objective function is as follows:

$$\max_{\Phi} I(\Phi(x), \Phi(x'))$$

Where $\Phi$ is the classification neural network, $x$ is the input, and $x'=g(x)$ is the transformation (random perturbation of the input) of $x$ (i.e. rotation, maximizing, minimizing, etc.). This is equivalent to maximizing the predictability of $\Phi(x)$ from $\Phi(x')$ and vice versa. The mutual information function is defined by:

$$I(X, Y) = \sum_{y \in Y} \sum_{x \in X} P_{X,Y}(x,y) \log \frac{P_{X,Y}(x,y)}{P_X(x)P_Y(y)}$$

(3)

The loss of invariant information clustering is therefore defined by:

$$L_{IIC} = -I(\Phi(x), \Phi(x'))$$

(4)

We adopt the IIC loss to the NLP domain by changing the nature of the random perturbation introduced to the input.

3.3.2 Match Loss

The output of the model’s last layer might be the same for all the different train sentences in some cases. To tackle this issue, we encourage the similarity between the last layer’s outputs $\Phi(x)$ and $\Phi(x')$ by adding a match loss. This loss is proportional to the cosine similarity between the two outputs and it is inspired from (Ansell et al., 2021) with the following:

$$L_M = -0.1 \sum \frac{\Phi(x) \cdot \Phi(x')}{\|\Phi(x)\| \|\Phi(x')\|}$$

(5)

3.4 Sense Embedding: Getting New Word Vectors

The architecture of our MIM model is very simple. It is formed of three projection layers with ReLU activation function. The final layer is equipped with the softmax function to get a probability distribution vector usable as an input to our loss. The hidden size of one linear layer is set to the double of RoBERTa\textsubscript{LARGE}’s and BERT\textsubscript{LARGE}’s hidden state size which is 1024.

For each target word, we train a model while providing the pairs of extracted representations belonging to the same cluster. In other terms, the target word’s representations from the original sentence and the sentence with lexical perturbation respectively.

The training concerns 8 runs over 5 epochs with a batch size of 32 using Adam optimizer (Kingma and Ba, 2014). The learning rate starts with 2e-5 and then is reduced linearly to zero over the remaining training time. The best model results from

\textsuperscript{1}https://docs.google.com/spreadsheets/d/1RK0vpslB98Nh_EOC4A2Byn8_201tmP0dpNwu4w7xI/edit?usp=sharing
the epoch minimizing the validation loss. The validation set is the pairs of sentences from the test dataset.

Once the training is complete, the hidden state representation of the first layer is extracted for each test word vector of the original sentence. Thus, the target word has a new projected representation.

### 3.5 Clustering

To cluster the instances into senses, we used the agglomerative clustering method. The same setup as in (Amrami and Goldberg, 2018, 2019) is used along with cosine distance and average linkage. To choose the number of clusters (senses) of each target word, we follow two approaches: (i) Fix the number of senses as in (Amrami and Goldberg, 2018, 2019; Ansell et al., 2021). (ii) Use a dynamic number of clusters based on its polysemy score obtained using the unsupervised word polysemy quantification (Xypolopoulos et al., 2020). For the dynamic clustering, we use the best configuration in the paper with dimensionality D equal to 3 and a level L equal to 8.

### 4 Evaluation

Several competitions were organized to systematically evaluate various methods applied for WSI, including SemEval-2007 task 02 (Agirre and Soroa, 2007), SemEval-2010 task 14 (Manandhar and Klapaftis, 2009) and SemEval-2013 task 13 (Jurgens and Klapaftis, 2013). The two tasks of SemEval-2010 and SemEval-2013 are considered as the benchmark for WSI. In this section, we publish and analyse the mean and standard deviation over 8 runs of the previously described model on the two mentioned tasks: SemEval-2010 task 14 and SemEval-2013 task 13.

#### 4.1 SemEval-2010 task 14:

On one hand, the primary objective of the SemEval-2010 WSI challenge is to compare unsupervised word-sense induction systems. It provides a mapping mechanism for evaluating WSI systems using the WSD dataset. The target word dataset consists of 100 tagged words, 50 nouns and 50 verbs extracted from OntoNet (Hovy et al., 2006). In the test set, each target word has around one hundred instances to be clustered. To learn its senses, a training set containing approximately 10,000 instances is provided for each target word. The training set is created using a semi-automatic web-based method.

For each sense of the target word in WordNet (Fellbaum, 1998), the query grabs all the sentences containing its corresponding stems and lemmas using Yahoo! search API. Each instance in the test dataset in this task is labeled with one sense only.

The performance in this task is measured with V-Measure (Rosenberg and Hirschberg, 2007) (biased toward high number of clusters) and F-Score (biased toward low number of clusters). We report the overall performance (AVG) defined as the geometric mean of these two metrics.

#### 4.2 SemEval-2013 task 13:

On the other hand, SemEval-2013 task 13 is a task for evaluating Word Sense Induction and Disambiguation systems in a context where instances are tagged with many senses whose applicability is weighted accordingly (Fuzzy Setting). The task focuses on disambiguating senses for 50 target lemmas: 20 nouns, 20 verbs, and 10 adjectives. The ukWac corpus (Baroni et al., 2009) is provided as a training corpus. It contains large number of instances crawled from the web and can be filtered by lemma, POS tag and many more filters. Test data are drawn from the Open American National Corpus (Ide and Suderman, 2004) across a variety of genres and from both the spoken and written portions of the corpus.

The performance in this task is measured with Fuzzy B-Cubed (F-BC) (Bagga and Baldwin, 1998). It is a generalized version of B-Cubed that deals with the fuzzy setting and Fuzzy Normalized Mutual Information (F-NMI). The latter is a generalized version of mutual information that deals with multi-sense annotation. We report as well the overall performance (AVG).

#### 4.3 Experiments

In order to prepare the training set of SemEval 2010 task 14, we chose randomly 3500 sentences from the provided training dataset of this task for each target word. For SemEval 2013 task 13, we extracted for each tagged target word up to 3500 random sentences from ukWac. Note that, if some of the target words in SemEval 2013 task 13 do not have 3500 sentences on ukWac, we extracted all the possible sentences. Following, we generate the paraphrases for both datasets by integrating the random perturbation described in section 3.1. The

---

2https://corpora.dipintra.it/public/run.cgi/first_form
Table 1: Comparison of WSI-specific techniques on SemEval 2010 task 14

| Method                | # Clusters | V-Measure | F-score  | AVG  |
|-----------------------|------------|-----------|----------|------|
| RoBERTa\textsubscript{LARGE} | 7          | 39.8      | 67.18    | 51.71 |
| RoBERTa\textsubscript{LARGE} (+MIM) | 7          | 44.8±0.10 | 67.74±0.78 | 55.1±0.88 |
| RoBERTa\textsubscript{LARGE} Dynamic | 7          | 40.1      | 65.23    | 51.14 |
| RoBERTa\textsubscript{LARGE} (+MIM) Dynamic | 7          | 42.3±0.45 | 67.39±0.37 | 53.28±0.35 |
| BERT\textsubscript{LARGE} | 7          | 40.34±0.88 | 64.75±1.08 | 51.14±0.93 |
| BERT\textsubscript{LARGE} (+MIM) | 7          | 41.2      | 67.42    | 52.6 |
| BERT\textsubscript{LARGE} Dynamic | 7          | 40.5      | 66.64    | 51.95 |
| DeBERTa\textsubscript{XLARGE} | 7          | 40.6      | 67.52    | 52.36 |
| DeBERTa\textsubscript{XLARGE} (+MIM) Dynamic | 7          | 40.59±1.23 | 68.05±0.76 | 52.49±1.1 |
| PolyLM\textsubscript{BASE} (Ansell et al., 2021) | 8          | 40.5      | 65.8     | 51.6  |
| PolyLM\textsubscript{SMALL} (Ansell et al., 2021) | 8          | 35.7      | 65.6     | 48.4  |
| Bert+DP (Amrami and Goldberg, 2019) | 7          | 40.4      | 71.3     | 53.6  |
| AutoSense (Amplayo et al., 2018) Dynamic | 9.8        | 61.7      | 24.59    | 22.16 |

Table 2: Comparison of WSI-specific techniques on SemEval 2013 task 13

| Method                | # Clusters | F-BC   | F-NMI   | AVG  |
|-----------------------|------------|--------|---------|------|
| RoBERTa\textsubscript{LARGE} | 7          | 64.1   | 19.28   | 35.16 |
| RoBERTa\textsubscript{LARGE} (+MIM) | 7          | 62.87±0.5 | 21.84±0.74 | 37.07±0.73 |
| RoBERTa\textsubscript{LARGE} Dynamic | 7          | 64.2   | 16.11   | 32.16 |
| RoBERTa\textsubscript{LARGE} (+MIM) Dynamic | 7          | 65.02±0.44 | 20.86±0.66 | 36.83±0.68 |
| BERT\textsubscript{LARGE} | 7          | 62.4   | 21.58   | 36.7  |
| BERT\textsubscript{LARGE} (+MIM) | 7          | 62.8±0.49 | 22.74±0.66 | 37.79±0.52 |
| BERT\textsubscript{LARGE} Dynamic | 7          | 63.16  | 18.57   | 34.25 |
| BERT\textsubscript{LARGE} (+MIM) Dynamic | 7          | 64.81  | 20.86   | 36.77 |
| DeBERTa\textsubscript{XLARGE} | 7          | 62.85±0.35 | 20.64±0.51 | 36.01±0.53 |
| DeBERTa\textsubscript{XLARGE} (+MIM) | 7          | 64.24  | 17.79   | 33.8  |
| DeBERTa\textsubscript{XLARGE} Dynamic | 7          | 64.64±0.39 | 19.67±0.58 | 35.66±0.6 |
| PolyLM\textsubscript{BASE} (Ansell et al., 2021) | 8          | 64.8   | 23.23   | 38.3  |
| PolyLM\textsubscript{SMALL} (Ansell et al., 2021) | 8          | 64.5   | 18.5    | 34.5  |
| Bert+DP (Amrami and Goldberg, 2019) | 7          | 64.5   | 21.4    | 37.1  |
| LSDP (Amrami and Goldberg, 2018) | 7          | 57.5   | 11.3    | 25.4  |
| AutoSense (Amplayo et al., 2018) Dynamic | 9.8        | 61.7   | 7.96    | 22.16 |

Table 3: The average perturbation percentage between the input text and the paraphrase. This percentage represents the proportion of changed unigrams.

| Dataset     | Train | Test |
|-------------|-------|------|
| SemEval-2010 Task 14 | 3.02% | 13.5% |
| SemEval-2013 Task 13 | 16.05% | 9.95% |

To infer the sense of a test instance in SemEval 2010, we first apply the agglomerative clustering method on the extracted RoBERTa\textsubscript{LARGE}, BERT\textsubscript{LARGE} and DeBERTa\textsubscript{XLARGE} vectors of the target word in the test instances (Section 3.2). The aforementioned step studies the effect of our word vectors enriching method. Second, for the model to be tested, we forward the test word vectors to the trained model and extract the corresponding hidden state of the first layer. This state is considered as the new word representation (sense embedding) of dimension 2048.

Finally, we applied agglomerative clustering on the new word representations implementing our clus-
tering solution. We assigned each instance to a single cluster.

The results of the evaluation on both \textit{SemEval-2010} and \textit{SemEval-2013} tasks are presented in tables 1 and 2 respectively providing the comparison with other WSI systems.

In the \textit{SemEval 2013} task, there is a possibility for a word to have multiple senses with a corresponding degree of applicability. Thus, once the agglomerative clustering applied, we convert the cosine similarity distances between each target word’s representation and the centroids of the different clusters to a vector of probabilities using the softmax function. These probabilities are considered as the senses’ degrees of applicability. The average number of clusters for each dataset in the dynamic setting is presented in table 5.

| Dataset | Average # of clusters |
|---------|------------------------|
| \textit{SemEval-2010 Task 14} | 6.73 |
| \textit{SemEval-2013 Task 13} | 5.36 |

Table 5: The average number of clusters obtained by using the polysemy scores on \textit{SemEval 2010} and \textit{SemEval 2013} test datasets

### 4.4 Results

Table 1 shows the performance of our approach in comparison to other baselines on \textit{SemEval 2010 task 14}. The best performing system, among the baselines, is BERT+DP (Amrami and Goldberg, 2019) providing the highest F-score of 71.3%.

With our method, RoBERTa\textsubscript{LARGE} outperforms all baselines in both settings: Fixed and dynamic number of clusters. This finding highlights the importance of our MIM approach that allows for an improvement of 1.5 absolute points over the previous state-of-the-art in terms of average score. In addition, we observe that the only model using dynamic clustering among the baselines (AutoSense) is largely outperformed by the other methods using a fixed number of clusters. However, given that WSI is an unsupervised task, the fixed number of clusters is supposed to be arbitrary and there is no guarantee that using the same number of clusters on other datasets would be optimal. Our proposed dynamic approach to choose the number of clusters did not deteriorate the performance of our method and in some cases led to a better performance (BERT\textsubscript{LARGE} for example).

\textit{SemEval 2013 task 13} performances are shown in Table 2. The best performing baseline is PolyLM\textsubscript{BASE} providing the highest F-BC and F-NMI scores. Although our approach did not outperform this baseline, it shows to be very competitive. In fact, the results on \textit{SemEval 2013 task 13}, shows again the positive contribution of our MIM approach, as we can observe a significant improvement whenever it is applied. For example, applying MIM to RoBERTa\textsubscript{LARGE} with dynamic clustering led to an increase of more than 4 absolute points in terms of average score.

To sum things up, (1) our proposed intermediate MIM phase led on average to an improved hierarchical clustering and (2) the dynamic approach to choose the number of clusters maintained the stable and competitive performance of our different evaluated models.

### 5 Best LM Layer

During the evaluation in section 4, we used the list provided by BERTScore (Zhang* et al., 2020) authors regarding the best performing layer. This choice is motivated by the fact that we are dealing with an unsupervised task, thus it is not possible to tune such a hyper-parameter without access to gold annotations. However, Zhang* et al. (2020) chose the best layer based on how good it performs in the task of machine translation evaluation. Dealing with a WSI task, there is no guarantee that the best layer is the same. Thus we carry out a study of the variation of the agglomerative clustering final score with respect to the layer used for the extraction of
the vector representations. This study can help researchers in future works to choose the appropriate layer when dealing with a similar unsupervised task.

Figure 3 shows the variation of agglomerative clustering performance in function of the depth of the chosen layer. Interestingly, we see that the variation of performance follows a similar pattern in SemEval-2010 and SemEval-2013 which can suggest a generalizable pattern over word sense induction datasets. Also, we can see that the pattern changes across different models. Despite having a similar architecture, the best layer depth in RoBERTa\textsubscript{LARGE} (layer 10) differs significantly with respect to that of BERT\textsubscript{LARGE} (layer 21). A future work should focus on this discrepancy and study the semantic information captured by each model’s layers. Table 4 presents the results regarding the best layer of each pretrained model on SemEval 2010 task 14 and SemEval 2013 task 13. The best performing pretrained contextual embeddings for both tasks is DeBERTa\textsubscript{XLARGE} with a score that outperforms the state-of-the-art methods.

6 Conclusion

In this work, we introduced an unsupervised method for the WSI task based on the tuning of contextual word embeddings extracted from a pretrained language model. The method generates paraphrases of the input sentences. Hence, both sentences belong to the same sense cluster. Next, it uses both sentences to train a MIM neural network that maximizes the mutual information between the two sentences’ outputs and minimizes the integrated match loss. The method improves on the state-of-the-art in one of the two WSI tasks.

We also use the polysemy score to test the dynamic number of senses setup as it claims superiority over the fixed setting in two out of six experiments. The MIM method proves, in most cases, an improvement in score while it does not deteriorate the performance in the others.

The extraction of representations for the target word depends on the chosen layer from the used pretrained language model. Thus, inspired by previous works, we conduct a comparison that helps the future studies in this choice.

Limitations

The aforementioned method presents an important improvement over some of the state-of-the-art solutions for WSI tasks. However, it suffers from some limitations that are worth highlighting:

(1) This method is training a MIM model from
As a consequence, our method suffers from higher words before applying agglomerative clustering. (2) Using the pretrained language models partially in our pipeline makes our method costly in terms of computation time when comparing with PolyLM. As consequence, our method suffers from higher number of parameters especially with models of bigger size such as DeBERTa. Thus, a further approach is to test with smaller models (i.e. DistilBERT) that could maintain the same good performance with faster training and inference time.

References

Eneko Agirre, Enrique Alfonseca, Keith Hall, Jana Straková, Marius Pasca, and Aitor Soroa. 2009. A study on similarity and relatedness using distributional and wordnet-based approaches. pages 19–27.

Eneko Agirre and Aitor Soroa. 2007. SemEval-2007 task 02: Evaluating word sense induction and discrimination systems. In Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007), pages 7–12, Prague, Czech Republic. Association for Computational Linguistics.

Reinald Kim Amplayo, Seung won Hwang, and Min Song. 2018. Autosense model for word sense induction.

Asaf Amrami and Yoav Goldberg. 2018. Word sense induction with neural biLM and symmetric patterns. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 4860–4867, Brussels, Belgium. Association for Computational Linguistics.

Asaf Amrami and Yoav Goldberg. 2019. Towards better substitution-based word sense induction.

Alan Ansell, Felipe Bravo-Marquez, and Bernhard Pfahringer. 2021. PolyLM: Learning about polysemy through language modeling. In Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, pages 563–574, Online. Association for Computational Linguistics.

Amit Bagga and Breck Baldwin. 1998. Entity-based cross-document coreferencing using the vector space model. In 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 1, pages 79–85, Montreal, Quebec, Canada. Association for Computational Linguistics.

Marco Baroni, Silvia Bernardini, Adriano Ferreira, and Eros Zanchetta. 2009. The wacky wide web: A collection of very large linguistically processed web-crawled corpora. Language Resources and Evaluation, 43:209–226.

Baobao Chang, Wenzhe Pei, and Miaohong Chen. 2014. Inducing word sense with automatically learned hidden concepts. In Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers, pages 355–364, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

Jr Corrêa and Diego Amancio. 2018. Word sense induction using word embeddings and community detection in complex networks. Physica A: Statistical Mechanics and its Applications, 523.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Ingo Feinerer and Kurt Hornik. 2020. wordnet: WordNet Interface. R package version 0.1-15.

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

Kartik Goyal and Eduard Hovy. 2014. Unsupervised word sense induction using distributional statistics. In Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers, pages 1302–1310, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: Decoding-enhanced bert with disentangled attention. In International Conference on Learning Representations.

Eduard Hovy, Mitchell Marcus, Martha Palmer, Lance Ramshaw, and Ralph Weischedel. 2006. Ontonotes: The 90

Nancy Ide and Keith Suderman. 2004. The American national corpus first release. In Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC’04), Lisbon, Portugal. European Language Resources Association (ELRA).

Xu Ji, João F. Henriques, and Andrea Vedaldi. 2019. Invariant information clustering for unsupervised image classification and segmentation.

David Jurgens and Ioannis Klapaftis. 2013. SemEval-2013 task 13: Word sense induction for graded and non-graded senses. In Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013), pages 290–299, Atlanta, Georgia, USA. Association for Computational Linguistics.
