Word sense induction with agglomerative clustering and mutual information maximization

Hadi Abdine, Moussa Kamal Eddine, Davide Buscaldi, Michalis Vazirgiannis

Word Sense Induction with Agglomerative Clustering and Mutual Information Maximization

Hadi Abdine[1], **Moussa Kamal Eddine**[1], **Davide Buscaldi**[3], **Michalis Vazirgiannis**[1,2]
[1]École Polytechnique, [2]AUEB, [3]Université Sorbonne Paris Nord

Abstract

Word sense induction (WSI) is a challenging 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. In contrast, others employ off-the-shelf pre-trained language models with additional strategies to induce senses. This paper proposes a novel unsupervised method based on hierarchical clustering and invariant information clustering (IIC). The IIC loss 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 show that our approach is at least on par with the state-of-the-art baselines, outperforming them in several configurations. The code and data to reproduce this work are available to the public.

1 Introduction

The automatic identification of a word's senses is an open problem in natural language processing, known as "word sense induction" (WSI). WSI is closely related to the word sense disambiguation task (WSD). While the latter relies on a predefined sense inventory (i.e., WordNet (Fellbaum, 1998; Wallace, 2007; Feinerer and Hornik, 2020)) and aims to classify the word's sense in context, the former focuses on clustering a collection of sentences according to the target word senses. For example, Figure 1 shows the different clusters obtained using our approach[1] on 3000 sentences that contain the word *bank* collected from Wikipedia. Note that the in this case, the senses and their number are not predefined, which highlights the difference between WSI and WSD.

Word senses are more beneficial than simple word forms for various tasks, including Information Retrieval and Machine Translation (Pantel and Lin, 2002). Word senses 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. Although the current state-of-the-art methods reasonably tackle this problem, they have significant limitations that should be addressed. For example, in their approaches, Ansell et al. (2021) and Amrami and Goldberg (2019) choose a fixed number of senses regardless of the target word without an explicit justification for their choices. On the other hand, Ansell et al. (2021) approach requires the pretraining of a new language model with a fixed vocabulary specific to the task. Applying their approach to a new vocabulary or a new language will be computationally expensive, which can impede the process of experimentation.

This paper includes the following contributions:

1) We propose a new unsupervised method leveraging pretrained language models, hierarchical clustering, and mutual information maximization. Our approach addresses some limitations of the previous efforts while providing a competitive performance.

2) We apply a new method to estimate a dynamic number of senses for target words. This method re-

[1]with RoBERTa$_{LARGE}$ (Liu et al., 2019) as underlying model

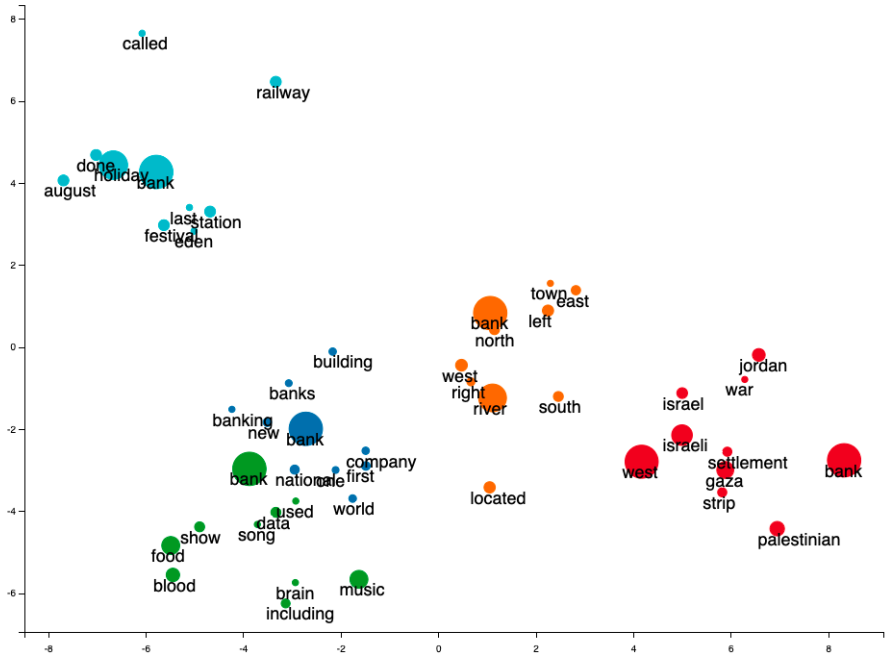

Figure 1: The different sense-based clusters of the word **bank** with the most frequent words used in the corresponding contexts. We use PCA to project the clusters' centroids to a 2D space. Each color corresponds to a cluster. The size of the points represents the frequency of the words in their corresponding cluster.

lies on word polysemy quantification (Xypolopoulos et al., 2021).

3) We study the variation of performance w.r.t the depth of the selected layer. Our findings in Section 5, covering four different models, are valuable for researchers conducting future work on WSI.

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., 2019), 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 seperating 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-

trained language models.
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 construct a synthetic dataset of pairs, each consisting of a sentence paired with a randomly perturbed version as explained in Section 3.1. Second, we extract the pair of hidden state representations of the target word using a pretrained language model (e.g., BERT). In our experiments, we consider three widely adopted language models: RoBERTa$_{LARGE}$, BERT$_{LARGE}$ and DeBERTa$^{mnli}_{XLARGE}$. Third, we train an MIM (Mutual Information Maximization) model where: (1) Considering an instance of the hidden state representations pairs, the network's is trains using two objectives: 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 smaller loss on the predefined test set. (3) We consider the output of the first layer as the new vector representation for the target word. Fourth, for each target word in the evaluation datasets, we apply the agglomerative clustering method on the new vector representations 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., 2021) of each target word.
The main steps are detailed in the following subsections.

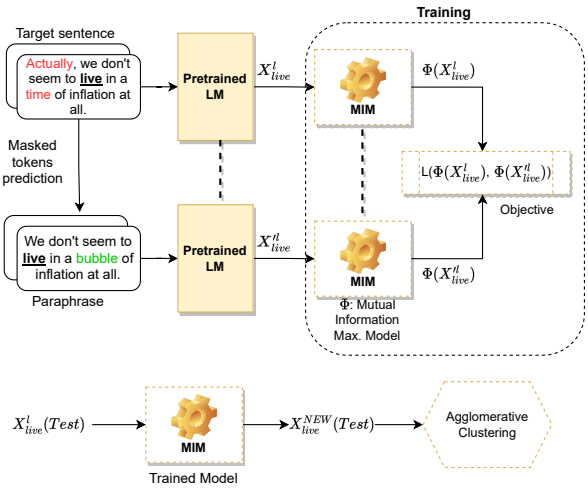

Figure 2: The pipeline of our method: For the word "live" chosen as target, a list of sentences is provided. BART is used to generate their corresponding paraphrases. The hidden representation X^l_{live} of the target word is extracted from the layer **l** of a pretrained language model. Dashed line 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 training a model to rebuild a corrupted version of the original sentences using an arbitrary noising function. It is based on a standard Tranformer-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 perturbed 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 it's perturbed 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 $BART_{BASE}$ 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$_{LARGE}$, RoBERTa$_{LARGE}$ and DeBERTa$_{XLARGE}$, we extracted representations of the target word from 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[2].

At this stage, if the target word is broken down into multiple tokens, we computed the average vector of the corresponding word pieces. Note that, while generating the perturbation on the input text using BART$_{BASE}$, 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:

$$L = L_{IIC} + L_M \tag{1}$$

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:

$$max_{\Phi} I(\Phi(x), \Phi(x')) \tag{2}$$

Where Φ 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:

$$I(X,Y) = \sum_{y \in Y} \sum_{x \in X} P_{X,Y}(x,y) log \frac{P_{X,Y}(x,y)}{P_X(x) P_Y(y)} \tag{3}$$

The loss of invariant information clustering is therefore defined by:

$$L_{IIC} = -I(\Phi(x), \Phi(x')) \tag{4}$$

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:

$$L_M = -0.1 \sum \frac{\Phi(x) \cdot \Phi(x')}{\|\Phi(x)\| \|\Phi(x')\|} \tag{5}$$

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$_{LARGE}$'s and BERT$_{LARGE}$'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, 2015). The learning rate starts with 2e-5 and then is reduced linearly to zero over the remaining training time. The best model results from the epoch minimizing the validation loss. The validation set represents 10% of pairs of sentences drawn

[2] https://docs.google.com/spreadsheets/d/1RKOVpselB98Nnh_EOC4A2BYn8_201tmPODpNWu4w7xI/edit?usp=sharing

randomly from the train 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., 2021). 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[3]. 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 average percentage of perturbation for each dataset is presented in table 3.

[3] https://corpora.dipintra.it/public/run.cgi/first_form

Model	# Clusters	V-Measure	F-score	AVG
RoBERTa$^{17}_{LARGE}$	7	39.8	67.18	51.71
RoBERTa$^{17}_{LARGE}$ (+MIM)	7	**46.26±0.51**	68.18±0.4	**56.16±0.42**
RoBERTa$^{17}_{LARGE}$	Dynamic	37	67.42	49.94
RoBERTa$^{17}_{LARGE}$ (+MIM)	Dynamic	45.06±0.92	68.79±0.33	55.67±0.54
BERT$^{18}_{LARGE}$	7	40.1	65.23	51.14
BERT$^{18}_{LARGE}$ (+MIM)	7	40.51±0.87	64.89±1.28	51.26±1.02
BERT$^{18}_{LARGE}$	Dynamic	41.2	67.17	52.6
BERT$^{18}_{LARGE}$ (+MIM)	Dynamic	41.8±0.49	67.43±0.36	53.1±0.4
DeBERTa$^{40}_{XLARGE}$	7	40.5	66.64	51.95
DeBERTa$^{40}_{XLARGE}$ (+MIM)	7	40.05±0.69	66.93±0.48	51.77±0.58
DeBERTa$^{40}_{XLARGE}$	Dynamic	40.6	67.52	52.36
DeBERTa$^{40}_{XLARGE}$ (+MIM)	Dynamic	40.58±0.92	67.89±0.55	52.48±0.76
PolyLM$_{BASE}$ (Ansell et al., 2021)	8	40.5	65.8	51.6
PolyLM$_{SMALL}$ (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., 2019)	Dynamic	9.8	61.7	24.59

Table 1: Evaluation of WSI models on SemEval 2010 task 14. The (+MIM) label indicates that the mutual information maximization is applyed to obtain the clustered vectors. Otherwise, the vectors from the pretrained langauge models are directly used.

Method	# Clusters	F-BC	F-NMI	AVG
RoBERTa$^{17}_{LARGE}$	7	64.1	19.28	35.16
RoBERTa$^{17}_{LARGE}$ (+MIM)	7	62.49±0.48	21.5±0.62	36.67±0.64
RoBERTa$^{17}_{LARGE}$	Dynamic	64.2	16.11	32.16
RoBERTa$^{17}_{LARGE}$ (+MIM)	Dynamic	64.8±0.29	19.95±0.63	35.95±0.56
BERT$^{18}_{LARGE}$	7	62.4	21.58	36.7
BERT$^{18}_{LARGE}$ (+MIM)	7	62.63±0.4	22.54±0.75	37.56±0.73
BERT$^{18}_{LARGE}$	Dynamic	64.81	20.86	36.77
BERT$^{18}_{LARGE}$ (+MIM)	Dynamic	64.42±0.31	21.22±0.59	36.97±0.54
DeBERTa$^{40}_{XLARGE}$	7	63.16	18.57	34.25
DeBERTa$^{40}_{XLARGE}$ (+MIM)	7	62.52±0.43	20.18±0.5	35.52±0.51
DeBERTa$^{40}_{XLARGE}$	Dynamic	64.24	17.79	33.8
DeBERTa$^{40}_{XLARGE}$ (+MIM)	Dynamic	64.44±0.48	19.27±0.4	35.26±0.32
PolyLM$_{BASE}$ (Ansell et al., 2021)	8	**64.8**	**23**	**38.3**
PolyLM$_{SMALL}$ (Ansell et al., 2021)	8	64.5	18.5	34.5
BERT+DP (Amrami and Goldberg, 2019)	7	64	21.4	37
LSDP (Amrami and Goldberg, 2018)	7	57.5	11.3	25.4
AutoSense (Amplayo et al., 2019)	Dynamic	61.7	7.96	22.16

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

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

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

The instances in *SemEval-2010 task 14* and *SemEval-2013 task 13* datasets contain some of the target words with morphological variability. Hence, lemmatizing is required to identify the target lemma during the vector extraction phase. Given this word and its POS tag, we use the WordNetLemmatizer from *NLTK* library to find its position inside both the sentence and its paraphrase followed by extracting the corresponding RoBERTa, BERT and DeBERTa vectors. These vectors are used to train the model as described earlier.

To infer the sense of a instance in *SemEval 2010*, we first apply the agglomerative clustering method on the extracted RoBERTa$_{LARGE}$, BERT$_{LARGE}$ and DeBERTa$_{XLARGE}$ vectors of the target word in the SemEval 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.

Model	#Clusters	SemEval-2010				SemEval-2013			
		Layer	V-measure	F-score	AVG	Layer	F-BC	F-NMI	AVG
RoBERTa$_{LARGE}$	7	10	43.6	68.12	54.5	9	63.87	23	38.32
RoBERTa$_{LARGE}$	dynamic	10	41.9	68.52	53.58	9	65.08	18.84	35.02
BERT$_{LARGE}$	7	21	40.8	66.7	52.17	20	63.16	22.07	37.34
BERT$_{LARGE}$	dynamic	21	41.3	67.65	52.85	20	65.54	21.26	37.32
DeBERTa$_{XLARGE}$	7	32	**49**	69.48	**58.35**	33	64.86	**24.14**	**39.57**
DeBERTa$_{XLARGE}$	dynamic	32	46.4	**69.49**	56.78	33	**66.62**	21.71	38.03

Table 4: The best layers of different pretrained language models on *SemEval-2010 Task 14* and *SemEval-2013 Task 13*

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

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

In the *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
SemEval-2010 Task 14	6.73
SemEval-2013 Task 13	5.36

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

4.4 Results

Table 1 shows the performance of our approach in comparison to other baselines on *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$_{LARGE}$ 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 2.5 absolute points over the previous state-of-the-art in terms on 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$^{18}_{LARGE}$ for example).

SemEval 2013 task 13 performances are shown in Table 2. The best performing baseline is PolyLM$_{BASE}$ 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 *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$_{LARGE}$ with dynamic clustering led to an increase of 3.8 absolute points in terms of average score. On the other hand, our method has two main advantages over PolyLM$_{BASE}$: (1) It can use the dynamic number of clusters compared to eight fixed senses for all words in PolyLM. (2) It does not require a computational-heavy pretraining to apply WSI on other languages. Indeed our method can be applied on other languages using already pretrained language models such as CamemBERT (Martin et al., 2020) or BARThez (Kamal Eddine et al., 2021) for the French language, AraBERT (Antoun et al., 2020) for the Arabic Language, etc..

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.

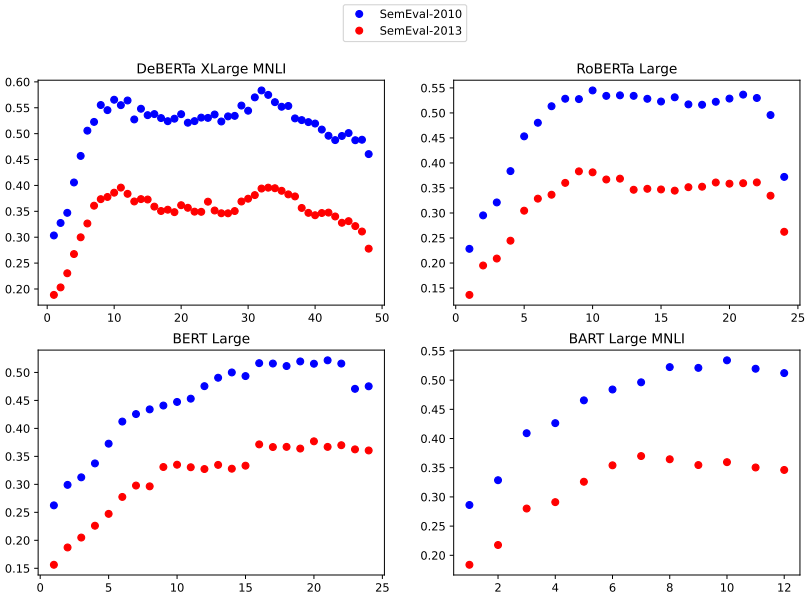

Figure 3: The AVG scores of SemEval-2010 and SemEval-2013 WSI tasks using agglomerative clustering on all the layers of different pretrained models.

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$_{LARGE}$ (layer 10) differs significantly with respect to that of BERT$_{LARGE}$ (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$_{XLARGE}$ 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 conducts a comparison that helps the future studies in this choice.

Limitations

The aforementioned method presents an important improvement over some of the-state-of-art solutions for WSI tasks. However, it suffers from some limitations that are worth highlighting:
(1) This method is training a MIM model from scratch for each target word proving a lack of generalizability. Thus, a further study can fulfil this task by training the MIM model starting from a pretrained language model for all target words. Applying this might yield to a general model that can give the sense embedding for all possible target 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 DitilBERT) that could maintain the same good performance with faster training and inference time. Finally, we must highlight the crucial role of the quality of the training data in determining the performance of our model on SemEval-2013 task 13. Unlike the comprehensive and meticulously constructed training sentences utilized in SemEval-2010 task 14, the training sentences sourced from ukWac for SemEval-2013 task 13 are characterized by their brevity, incompleteness, and nonuniform extraction from the web. To illustrate the disparities between the training sets for both tasks, we provide examples in the appendix.

References

Eneko Agirre, Enrique Alfonseca, Keith Hall, Jana Kravalova, Marius Paşca, and Aitor Soroa. 2009. A study on similarity and relatedness using distributional and wordnet-based approaches. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, NAACL '09, page 19–27, USA. Association for Computational Linguistics.

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. 2019. Autosense model for word sense induction. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):6212–6219.

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.

Wissam Antoun, Fady Baly, and Hazem Hajj. 2020. AraBERT: Transformer-based model for Arabic language understanding. In *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*, pages 9–15, Marseille, France. European Language Resource Association.

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 Ferraresi, 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. 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.

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.

Eduard H. Hovy, Mitchell P. Marcus, Martha Palmer, Lance A. Ramshaw, and Ralph M. Weischedel. 2006. Ontonotes: The 90% solution. In *HLT-NAACL*. The Association for Computational Linguistics.

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. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9865–9874.

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.

Moussa Kamal Eddine, Antoine Tixier, and Michalis Vazirgiannis. 2021. BARThez: a skilled pretrained French sequence-to-sequence model. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9369–9390, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Diederik Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, San Diega, CA, USA.

Alexandros Komninos and Suresh Manandhar. 2016. Dependency based embeddings for sentence classification tasks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1490–1500, San Diego, California. Association for Computational Linguistics.

Jey Han Lau, Paul Cook, Diana McCarthy, David Newman, and Timothy Baldwin. 2012. Word sense induction for novel sense detection. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 591–601, Avignon, France. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

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.

Suresh Manandhar and Ioannis Klapaftis. 2009. SemEval-2010 task 14: Evaluation setting for word sense induction & disambiguation systems. In *Proceedings of the Workshop on Semantic Evaluations: Recent Achievements and Future Directions (SEW-2009)*, pages 117–122, Boulder, Colorado. Association for Computational Linguistics.

Louis Martin, Benjamin Muller, Pedro Javier Ortiz Suárez, Yoann Dupont, Laurent Romary, Éric de la Clergerie, Djamé Seddah, and Benoît Sagot. 2020. CamemBERT: a tasty French language model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7203–7219, Online. Association for Computational Linguistics.

Tomás Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*.

Patrick Pantel and Dekang Lin. 2002. Discovering word senses from text. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '02, page 613–619, New York, NY, USA. Association for Computing Machinery.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Alec Radford and Karthik Narasimhan. 2018. Improving language understanding by generative pre-training.

Andrew Rosenberg and Julia Hirschberg. 2007. V-measure: A conditional entropy-based external cluster evaluation measure. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 410–420, Prague, Czech Republic. Association for Computational Linguistics.

Linfeng Song, Zhiguo Wang, Haitao Mi, and Daniel Gildea. 2016. Sense embedding learning for word sense induction. In *Proceedings of the Fifth Joint Conference on Lexical and Computational Semantics*, pages 85–90, Berlin, Germany. Association for Computational Linguistics.

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

Jean Véronis. 2004. Hyperlex: Lexical cartography for information retrieval. *Computer Speech & Language*, 18:223–252.

Mike Wallace. 2007. *Jawbone Java WordNet API*.

Jing Wang, Mohit Bansal, Kevin Gimpel, Brian D. Ziebart, and Clement T. Yu. 2015. A sense-topic model for word sense induction with unsupervised data enrichment. *Transactions of the Association for Computational Linguistics*, 3:59–71.

Christos Xypolopoulos, Antoine Tixier, and Michalis Vazirgiannis. 2021. Unsupervised word polysemy quantification with multiresolution grids of contextual embeddings. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3391–3401, Online. Association for Computational Linguistics.

Tianyi Zhang*, Varsha Kishore*, Felix Wu*, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

A Appendix

In what follows, we provide four examples from each SemEval WSI train and test datasets.

The Commission seeks comment on whether the analytical framework that was used to streamline AT &T 's services should be applied to incumbent LEC **access** services. In particular , the Commission seeks comment on which of the factors that it used in examining AT &T 's pricing behavior could be used to determine when to remove incumbent LEC **access** services from price cap regulation. It cites demand elasticity , supply elasticity , market share , and the pricing of services under price cap regulation as relevant factors .
This works fine if AudioPlayer is n't going to be subclassed. But what if you were going to create a class called StereoAudioPlayer that is a subclass of AudioPlayer ? This class would want **access** to the openSpeaker ( ) method so that it can override it and provide stereo-specific speaker initialization. You still do n't want the method generally available to random objects (and so it should n't be public) , but you want the subclass to have **access** to it-so protected is just the solution .
502.4 Floor or Ground Surfaces. Parking spaces and **access** aisles serving them shall comply with 302. **Access** aisles shall be at the same level as the parking spaces they serve. Changes in level are not permitted .
When developing kernel code , it is usually important to consider constraints and requirements of architectures other than your own. Otherwise , your code may not be portable to other architectures , as I recently discovered when an unaligned memory access bug was reported in a driver which I develop. Not having much familiarity with the concepts of unaligned memory **access** , I set out to research the topic and complete my understanding of the issues .

Table 6: Random examples for the target word 'Access' from SemEval-2010 task 14 training set

Baby Welcome to my eBay Shop. Please **add** me to your list of favourite sellers and
digital jesters guys said they would NEVER **add** collision detection to TM , as this is
Also in the Spanish version, but more were **added** especially for the Japanese Complete Editions
destination that you have entered . You can **add** any number of intermediate waypoints to

Table 7: Random examples for the target word 'Add' from SemEval-2013 task 13 training set

In more than four years , 2.2 billion yuan has been invested in the construction of harbors and docks , storage fields , support facilities and infrastructure of the ports and city , creating good conditions for building **access** to the sea for the Great Southwest .
The FDA is expected to approve today a program granting **access** free of charge to the drug AZT for children with AIDS .
Federal health officials are expected today to approve a program granting long - deferred **access** to the drug AZT for children with acquired immune deficiency syndrome .
The dispute stems from pretrial maneuvering in the pending court case , in which prosecutors have been demanding **access** to a host of internal company memos , reports and documents .

Table 8: Random examples for the target word 'Access' from SemEval-2010 task 14 test set

Lewinsky wrote "Return to Sender" on the envelope, **adding**, "You must be morons to send me this letter!"
For instance, the Post also has the story about the woman meeting with Clinton just days before his first Inaugural, but **adds** the detail that she says all the encounters were innocent.
if you **add** the um uh people of various sexual persuasions and those who never intend to marry and those who are retired and those who are um just looking for fun they people with families turn out to be such a small minority that they can't get the tax bill passed no matter what happens
The tripe with onions and garlic is cooked for several hours, posole or hominy is **added**, along with red chile.

Table 9: Random examples for the target word 'Add' from SemEval-2013 task 13 test set