Can Pre-trained Language Models Understand Definitions?

Stanford CS224N Custom Project

Tina Li, Xiaoyuan Ni, Xinran Zhao
Department of Computer Science
Stanford University
tinally, xniac, xzhaoar@stanford.edu

Abstract

Despite the great performance improvement on various tasks with pre-trained language models (PTLMs) in recent years, it remains unclear how much the models understand human languages. Motivated by the psycholinguistic fact that humans show great capability in concluding and describing concepts, we propose to examine the PTLMs' ability to understand the definitions with a novel generalized word-sense matching task. Through proposing a novel task, word sense matching (WSM), we require PTLM-based models to find the alignment between embedding space of word-in-context and descriptive sentences capture the word senses. With extensive experiments, we further show that PTLMs can understand definitions through finding the semantic relevance between the contexts and definitions and fail when the negative samples have similar contexts or the definitions are abstract.

1 Key Information to include

- Mentor: Christopher D. Manning
- External collaborators/External mentor/Sharing project: No

2 Introduction

Recently, pre-trained language models (PTLMs) have demonstrated significant improvements over multiple downstream natural language classification and generation tasks [1, 2, 3]. However, it remains unclear to what extent the models capture the training data distribution and understand the languages. Besides writing sentences or answering questions, which have been largely explored using PTLMs, humans can also give definitions to words and concepts from what they learn in daily life. Motivated by this, in this work, we aim to examine the PTLMs' ability to find correct definitions for words in context. In the area of natural language processing (NLP), the model ability to identify senses has long been explored by the classic task word sense disambiguation (WSD) [4]. WSD is commonly formed as an entity linking task that requires models to choose the correct sense of a word in its context from multiple choices defined in a given knowledge base as the sense inventory (e.g., WordNet [5])). Despite the great improvement in the recent years in the community [6, 7, 8, 9], there are some potential limitations of such task formation:

1. the dynamic semantics of words may not be captured precisely by pre-defined explicit senses;
2. models are only required to choose from a set of senses for each word locally and senses from other words, which can form global disambiguation, are excluded;
3. the multiple-choice format relying on a single knowledge base limits the potential downstream or real-world application. For example, the models can not handle the case when we have a descriptive sentence and intend to find its potential target from our whole vocabulary, where the

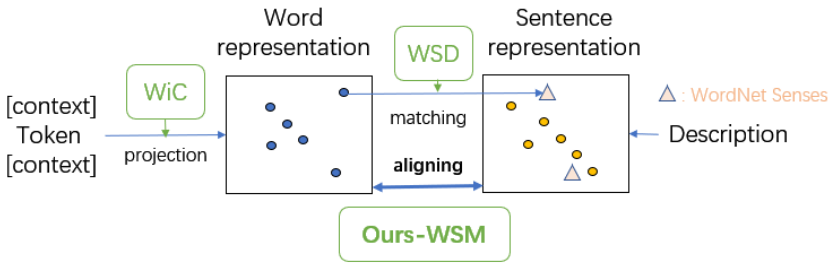

Figure 1: The differences between WiC, WSD, and our task. WiC studies how to project tokens in context to a representation space of word senses. WSD studies matching the word representation to a particular set of senses defined in a provided inventory, such as WordNet. In contrast, in our word sense matching task, we aim to align the space of word senses and descriptive sentences.

case can be commonly observed when searching items on e-commerce platforms or detecting objects from outputs of video captioning models.

Various literature emphasizes the importance of dynamic word senses regarding contexts. [10] has shown a strong effect of sense sharing for polysemous words in the same context. To capture implicit dynamic word senses related to the context, corresponding datasets and models are proposed [11, 12]. More recently, WiC [13] is proposed to ask models to compare if a target word shows similar senses in two sentences. In this project, we intend to further align the dynamic senses with their potential descriptive sentences. We propose a novel task, word sense matching (WSM), which requires the models to decide if a candidate sentence can correctly describe a word in context. WSM provides multiple interesting characteristics: (1) the descriptive sentences are collected from multiple sources and not limited by a single sense inventory; (2) as a binary classification task with three factors: word, context, description, it is suitable for evaluating various objectives (e.g., contextualized word representation and sentence representation) and methods. The difference in focus among WSD, WiC and our task is shown in Figure 1. WiC focuses on finding the correct projection from token to embedding space representing dynamic semantics, WSD focuses on matching the word embedding space to the embedding space of some particular pre-defined senses, and our WSM focuses on finding the general alignment between word representation and descriptive sentences.

Upon the creation of the task, we conduct extensive experiments to probe the PTLMs' capability in matching words with their sense descriptions through various different approaches and settings (details in Section 5). We find that: (1) null prompts with all parameters tuned is the best way to conduct prompt-based baselines; (2) PTLMs can solve the task reasonably well by forming it as semantic similarity task (referred to as *disjoint* in the later sections); (3) definitions from words' antonyms and abstract definitions are hard to be understood by PLTMs.

3 Related Work

3.1 Representation Learning

Learning the representation of languages has been an important task in NLP. Replacing the one-hot vector representation, word embedding methods [14, 15] have gained huge success through representing the tokens by their context. Recently, various contextualized pre-trained language representation models [16, 1, 2, 17] are proposed and further boost the model performance over various downstream. Dynamic word semantics is captured in the representation generated from these models since the token representation varies as the context differs. Besides token-level (or sub-token-level) embedding, entity embedding and sentence embedding have also been widely studied, with methods beyond simply averaging the embeddings for each token. For example, [18, 19] are proposed to represent the entities in probabilistic distribution to capture the dynamic semantic relations among entities. On the other hand, [20, 21] are proposed to acquire improved sentence representation by learning the sentence-level relation (such as semantic textual similarity or inference) in a contrastive learning manner. The essence of our proposed WSM can be considered as finding the alignment of two kinds of representation: the representation of words in context and the representation of

Split	Instance	Avg. Context Length	Avg. Definition Length	# Adjective	# Noun	# Verbs
Training	6,240	5.91	7.78	2,698	1,584	1,364
Validation	780	6.04	7.92	332	228	162
Test	780	7.93	9.76	168	294	278

Table 1: Statistics of different splits of WSM. #X denotes the number of X in the split.

Label	target	context	definition
1	bother	He is a bit of a bother.	Something or someone that causes trouble.
1	bank	Who do you bank with?	To keep your money in a particular bank.
0	bank	There are steep banks of snow.	An organization providing financial services.
0	perceivable	It is perceivable through the mist.	The act of looking for someone or something.

Table 2: Sample positive and negative examples from the dataset.

descriptive sentences. The projection from raw languages to the representation space is largely related to the representation learning methods introduced above.

3.2 Word Sense Disambiguation

The origin of the task of word sense disambiguation (WSD) can be sourced from [22], where the use of word senses in machine translation is recognized. Through the years, various related datasets and tasks are proposed [23, 24, 25, 26]. Conventionally, the task is formulated as a entity linking task where the models are required to choose a sense for each target word from its possible senses contained in a pre-defined sense inventory. With the rise of PTLMs, performance has been largely boosted with exploration on different ways to utilize them, including treating the sense definition as apposition [7], adding masked-sense prediction task [27], and directly using the hidden layers from PTLMs as the representation [8, 9]. In this work, we propose a novel task word sense matching that focuses on a different aspect of understanding definitions. We relieve the reliance on explicit sense labels from the inventories and ask the models to generally decide if a sentence can precisely describe the sense of a word in context.

4 Approach

Specifically, we aim to extend the classic word sense disambiguation task [4] from classifying the senses for every single word to general word-sense matching over multiple senses and words. In detail, the binary classification task is defined as follows: given a target word w in a context sentence c, the models are required to detect if a sentence g can be considered as a description to the definition of w and gives a prediction 1 (correct) or 0 (wrong).

The following part of the section will be organized as follows: first, we introduce the creation of the dataset for the task, then, we will introduce the motivation and design of proposed methods and baselines. Since it is a newly proposed task, most of the baselines will all be written from scratch, with the help of Hugginface Transformers [28]. The use of other models and techniques will also be specified in the following subsections.

4.1 Dataset Creation

The closest previous task with ours is the classic WSD [4, 29], where the models are required to distinguish the definitions of the words in context from a limited set of senses from common knowledge bases (e.g., WordNet). To further capture the dynamic nature of word semantics, we follow two guidelines to create our WSM dataset: (1) multiple sources: we do not rely on the sense definitions from any particular source, since the meaning of the target word can be dynamic and does not yield to any explicit sense name or definition. We extract the word-definition pairs from multiple sources including previous human annotations (i.e., SemCor [25]), which has widely been used in the community, WordNet [5], and commonly used dictionaries (e.g., Oxford, Webster, and Cambridge dictionaries); (2) multiple samples: to sample the negative examples, we consider both the local (senses of target words) and global samples (senses of other words). Besides, to further analyze how

Parameters	Manual	Trigger	Sep	Null
Bitfit	-	53.59%	57.81%	54.62%
All	50.32%	61.92%	60.13%	71.41%

Table 3: Performance on our WSM task with different prompt designs on the validation set with 32 examples per class. *Manual, Trigger, Sep*, and *Null* denote the performance (accuracy) with corresponding prompt pattern design introduced in Section 4.2. *Bitfit* and *All* denote training either the bias terms only or all parameters in the PTLMs.

PTLMs understand the general word sense matching, we tag the target words of the examples with the Part-of-Speech tags and report these categorical results in the later sections. At this stage, the dataset is based on English. Possible extensions to other languages can be implemented since most languages have their own dictionaries providing definitions and examples.

Statistics and examples of the created dataset are presented in Table 1 and Table 2, respectively. Upon the creation of the dataset, we check the effectiveness of the dataset with human evaluation. We invite three college students who speak English fluently to do a sampled sub-task with 50 questions. The average inter-annotator agreement (IAA) of their annotations is 92.67%, which suggests that humans can finish the task with good agreement.

4.2 Methodology and Baselines

As a newly proposed task, we intend to explore and compare the performance of various models on WSM. We show the ablation study for different design choices to inspire future research so that these models can be considered both as our methods and baselines. Besides ablation, we also implement an N-gram matching based Lesk Approach [30, 31] as the baseline, where the score of a definition is given by its n-gram coverage over the word context. The design choices are compared as follows:

1. ***Disjoint vs. joint:*** since our task can be considered as an extension of the original WSD, following its state-of-the-art models [9, 8, 32], the first set of baselines can be designed as bi-encoder-based *disjoint* models. Specifically, for each question, we use PTLMs to encode the context sentence and the definition (i.e., the gloss in WSD) as W and G , respectively, with average pooling on the embedding of each token. The score of the definition can then be extracted as the cosine similarity of W and G. We explore both BERT [1] and RoBERTa [2] as the backbone.

   Unlike the multiple-choice setting in the original WSD, WSM is formed as a binary classification problem, so that we are able to explore the use of prompt-based tuning that shows good performance over various tasks [33, 34, 35]. Specifically, we organize the factors (word, context, and definition) for each example into a single meaningful sentence (i.e., prompt pattern) and then conduct the prediction. Since only a single sentence will be passed to the PTLMs, we denote this line of baselines as a *joint* approach. An example of the organization is as follows "In <*context*>, the <*target*> <*label word*> be described as <*definition*>", where the label word is can or cannot for the label 1 and 0, respectively. The prediction can then be given by training a classifier with the sentence representation from PTLMs as features. Similarly, we experiment on BERT and RoBERTa with different sizes.

   One reason to distinguish disjoint and joint models is the consideration of time complexity in the real-world application. Suppose that we deployed the model and receive M queries (descriptions), with N candidate targets, the computation time of disjoint and joint models will be $O(M + N)$ and $O(MN)$, respectively (if constant time is required to feed one sentence to the PTLMs).

2. ***Prompt Design:*** Following the literature in prompt-based tuning, three factors are important towards its success: the pattern (manually designed or from generative models [34]), the verbalizer [1], and parameters to tune (all PTLM parameters, embedding layers only [36], or bias only [37]). Specifically, we explore four kinds of patterns (manual, trigger, sep, null) and two kinds of parameters to train (all parameters and bias only[2]). The details of the patterns are as follows: (1) manual: we manually create a sentence to link the *context*, *word*, and *definition*; (2) trigger: we

[1] A module converting original labels to meaningful words, such as 1 to *happy* for sentiment classification.

[2] This technique is known as Bitfit [37], which has been shown to be similarly effective as fine-tuning the whole PTLMs in previous work on prompt-based tuning [33]

put triggers as soft words between *context*, *word*, and *definition* and tune the embeddings of these triggers; (3) sep: we put a <sep> token between each pair of *context*, *word*, and *definition*; (4) null: following [33, 38], we simply form the prompt by joining *word*, *definition* , and *context* with blanks in-between. For the experiment on prompt design, we uniformly use RoBERTa-large as the backbone PTLM. To guide our later experiment, the results on the validation set are presented in Table 1. We could observe that fine-tuning on all parameters generally outperforms the Bitfit technique. One possible reason is that the task is comparatively harder than the normal few-shot learnable task, such as sentiment classification [39], and more training steps are needed. On the other hand, we could observe that the best strategy is using no extra words for the pattern at all. The reason behind this can be that the relations between word, definition, and context can already be learned implicitly with attention layers if we have enough parameters and additional tokens introduce extra noise.

3. ***Zero-shot, Few-shot, and Full Fine-tuning:*** The recent progress in PTLMs has led to the advance of learning a task with only a few examples [16, 40, 41, 42] (i.e., Few-shot Learning) to overcome the challenges in solving tasks that are not rich in data. Label collection can be extensively expensive for sense annotation (often requires experts to distinguish and annotate the detailed senses [13]). In some particular domains, such as when we try to match terms and definitions from medical reports, the data scarcity further increases. Motivated by this, we compare the model behavior with different sizes of training data: no fine-tuning example (zero-shot), 32 examples per label class (few-shot), and using the whole training data defined in Section 4. For zero-shot approaches, we explore the n-gram based Lesk-style approach, zero-shot classification of a large-scale pretrained language model (GPT-J [43]). In addition, we utilize cosine similarity of representation generated from PTLMs as another zero-shot approach by using the median of all calculated similarity scores in the validation set as the threshold.

5 Experiments

5.1 Experimental Setup

1. ***Data and Evaluation:*** The data preparation step has been discussed in Section 4. In the zero-shot setting, we directly apply the models to our test set. In the few-shot learning, we generate a few-shot training set with 32 examples per class and report the performance on the test set. In the full fine-tuning setting, we use the whole training set introduced previously.

   As for the evaluation method, since WSM is formatted as a binary classification task, we report the accuracy for the overall dataset and subset for each category (e.g., noun, verb).

2. ***Experimental details:*** For the model fine-tuning, we set the learning rate as 1e-5, batch size as 2, and epochs as 30. For other model configurations, we use the default ones from the original packages or repositories. To run the experiment, we use one NVIDIA TITAN X with 8 G ram. Extracting embeddings from PTLMs typically takes 30 seconds per 100 examples. The zero-shot evaluation with GPT-J typically takes 20 minutes per 100 examples. Other required steps (e.g., generating categorical performance report) take less than 10 seconds.

3. ***Baseline details:*** In Section 4.2, we have introduced the motivation and ideas behind the baselines to be included in comparison. In this section, we introduce some necessary implementation details of the baseline. For ***disjoint*** models, we separately encode the contexts and the definitions with "<*target*>: <*context*>" and "<*target*>: <*definition*>" as the raw inputs. For ***joint*** models, following the findings from Table 1, we use "<*target*> <*definition*> <*context*>" (i.e., null prompts) as the inputs. Considering its great success on various downstream tasks, we acquire pre-trained weights of BERT and RoBERTa from the contrastive learning based sentence embedding method, SimCSE [21]. If not specified, BERT, RoBERTa denote the supervised SimCSE models using them as the backbones. We use SimCSE-unsup to denote the unsupervised SimCSE model with RoBRETa-large as the backbone. More detailed comparison of these pre-trained weights can be found in the original github repository [3]. We also follow the original work to use Logistic Regression as the classification head with the inputs as the cosine similarity of the embeddings (for ***disjoint***) and representation of the prompt generated by SimCSE models (for ***joint***) .

[3] https://github.com/princeton-nlp/SimCSE

Method	Source						Part-of-Speech				Overall
	WN (400)	Collins (56)	Longman (90)	Webster (78)	Oxford (90)	Cambridge (66)	adj (204)	noun (268)	verb (254)	other (54)	(780)
Zero-shot											
Lesk	55.5%	71.43%	73.33%	67.95%	74.44%	72.73%	55.95%	63.64%	69.92%	52.63%	63.59%
GPT-J	49.75%	57.14%	61.11%	62.82%	61.11%	56.06%	45.55%	57.50%	57.66%	56.52%	54.74%
SimCSE-unsup	77.75%	80.36%	90.00%	85.90%	74.44%	80.30%	80.95%	79.87%	79.70%	78.95%	**80.00%**
Few-shot											
joint - BB	54.75%	55.36%	57.78%	57.69%	51.11%	53.03%	52.98%	57.14%	52.63%	60.53%	54.87%
joint- BL	52.75%	62.50%	67.78%	71.79%	61.11%	65.15%	54.90%	61.19%	62.99%	46.30%	59.10%
joint- RB	52.50%	50.00%	52.22%	56.41%	47.78%	54.55%	52.98%	54.22%	50.75%	44.74%	52.31%
joint- RL	56.00%	55.36%	55.56%	51.28%	44.44%	51.52%	55.95%	53.57%	51.88%	57.89%	53.72%
disjoint- BB	71.25%	69.64%	76.67%	82.05%	67.78%	66.67%	73.21%	73.38%	70.30%	68.42%	72.05%
disjoint- BL	71.75%	82.14%	90.00%	79.49%	76.67%	74.24%	70.83%	76.95%	79.70%	68.42%	76.15%
disjoint- RB	75.25%	80.36%	90.00%	85.90%	70.00%	69.70%	75.60%	79.87%	75.94%	73.68%	77.31%
disjoint- RL	74.75%	69.64%	85.56%	83.33%	73.33%	77.27%	74.40%	77.60%	77.82%	68.42%	76.54%
Full Fine-tuning											
joint - BB	73.25%	73.21%	75.56%	82.05%	74.44%	74.24%	70.83%	74.35%	77.44%	73.68%	74.62%
joint- BL	75.25%	73.21%	83.33%	79.49%	64.44%	72.73%	76.79%	74.35%	73.31%	84.21%	75.00%
joint- RB	69.50%	76.79%	73.33%	73.08%	68.89%	72.73%	69.05%	69.16%	74.06%	73.68%	71.03%
joint- RL	70.50%	73.21%	72.22%	66.67%	71.11%	71.21%	76.19%	67.53%	71.05%	68.42%	70.64%
disjoint- BB	73.25%	73.21%	75.56%	82.05%	74.44%	74.24%	70.83%	74.35%	77.44%	73.68%	74.62%
disjoint- BL	71.75%	78.57%	86.67%	83.33%	73.33%	74.24%	70.83%	75.97%	78.57%	71.05%	75.51%
disjoint- RB	75.50%	78.57%	90.00%	85.90%	68.89%	69.70%	76.19%	79.55%	75.56%	73.68%	77.18%
disjoint- RL	75.00%	67.86%	85.56%	84.62%	72.22%	75.76%	74.40%	78.25%	76.69%	68.42%	76.41%

Table 4: The statistics and performances of different methods over our test set. WN denotes that the example sentences come from WordNet. The rest columns under *source* denote the corresponding dictionary. *other* under POS denotes other part-of-speech tags including adverb, phrase, etc. The numbers in bracket denote the numbers of examples of the corresponding columns. *disjoint* and *joint* denote whether we encode the contexts and definitions separately, as described in Section 4.2. *BB, BL, RB, RL* denote the use of BERT-base, BERT-large, RoBERTa-base, RoBERTa-large as the backbone PTLMs, respectively.

5.2 Results and Analysis

The results are presented in Table 4. We could observe that:

1. ***Zero-shot learning works well:*** From Table 4, we could observe that some simple zero-shot baselines achieve good performance on our dataset. For example, n-gram overlapping-based Lesk approach achieves better performance than most few-shot joint models. On the other hand, surprisingly the unsupervised SimCSE model gets the best performance over all models.

   The reason can be that the unsupervised SimCSE sees more data (more than 1 million words) than the supervised ones and shows better capability in linking deeply related sentences. Similarly, in the original paper, we could observe that the unsupervised models outperform the supervised models on the text retrieval task TREC [44], which is formed as retrieving the entities from question describing them [4]. The success of the zero-shot models suggests the essence of our task as finding the relevance of heterogeneous sentences. However, we could also observe that GPT-J does not perform well with a joint prompt, which suggests that the inductive bias embedded in the PTLMs is not enough to find the complex relevance without further fine-tuning.

2. ***Joint models only work after fine-tuning:*** Comparing rows under few-shot and full fine-tuning, we can observe that with around 100 times more data, ***Joint*** models get a significant performance boost for 15-20 % accuracy. With only a few examples, though still perform better than a 50% accuracy majority baseline, ***Joint*** models fall far behind even the simplest Lesk model. The gap between joint models and the best-performing models largely shrinks with the large training set, which suggests that the inductive bias in the PTLMs is not necessary to give the prediction of WSM directly in a joint format so that training is necessary in this case.

3. ***Fine-tuning is not necessary for disjoint models:*** Similarly, if we compare rows with disjoint models, we can observe that the model performance for few-shot models is very close and even better than fully fine-tuned models. The performance differences mainly come from different choices of the backbone PTLMs, where the best performance comes from RoBERT-base for both

[4] An example is to find the location with the question: where is the oldest living thing on earth?

Label	Pred.	Context	Definition
0	0	It aroused the *tiger* in me	Time for Earth to make a complete rotation.
1	1	Black smoke *coiling* up into the sky	To wind or move in a spiral course.
0	1	You cannot *compartmentalize* your life like this!	Make uniform.
1	0	*Minor back roads*	Of lesser importance or stature or rank

Table 5: Correct and wrong predictions from the best-performing unsupervised SimCSE model (target words in *italics*). Pred. denotes the predicted labels.

Target	Emphasize
Context	You *emphasized* the high prevalence of mental illness and alcoholism.
WN Labels	to stress, single out as important (correct) ; Give extra weight to (a communication) (wrong)

Table 6: An example of the standardized WSD dataset [25], WN refers to WordNet, which contains the sense labels and corresponding definitions.

training size settings. The intuition behind this can be that, in a disjoint setting, the essential objective of our task as finding the deep semantic relation of two kinds of text can be nicely fulfilled already by the PTLMs. With only a small size of examples, the disjoint models are already specialized enough and a larger size of data may lead to over-fitting and causes a slight performance drop.

4. ***Performance differs on data from different sources:*** From the table we can observe that there is a clear gap between different columns representing test examples from different sources, for example, examples from Longman and Merriam-Webster dictionaries are generally easier to be solved than those from Oxford and Cambridge dictionaries. Though fine-tuning usually leads to performance boost over data from multiple sources, the consistent gaps among the columns still suggest that the potential styles of writing the definitions can cause a difference of around 15% accuracy for the best-performing models. However, the original WSD usually only collects the senses from a single sense inventory, which may limit the performance in real-world application as the styles of writing descriptive sentences can differ a lot. The comparison of the columns suggests that we should include the performance on different test subsets from different sources to clearly reveal the real performance of the models under different scenarios.

6 Analysis

6.1 Qualitative Study

We have revealed some interesting findings in Section 5.2 when discussing the results. The most surprising fact is that, the zero-shot unsupervised SimCSE model, which is pre-trained with self-supervised signals, performs the best among all the included models. To further understand the reason behind this, we conduct a qualitative evaluation on the correct and wrong predictions as presented in Table 5. From the first examples where both the label and prediction are 0, we could observe that the model has a good understanding of finding unrelated sentences. In the second row of the examples, we find that the model can surprisingly capture sentences that are semantically connected (e.g., *sky* and *wind*). For the wrong predictions, the third row shows an example that the model can not distinguish one word from the sense of its antonym. The reason can be that words and their antonyms can often be used in similar contexts. The final example shows that the model fails to capture the relation between abstract and concrete objects.

The above examples show that, despite the great performance achieved by aligning the contexts, some potential hard cases remained challenging: antonym senses and abstract definitions, which can motivate future work in creating a more challenging dataset.

6.2 Application to Original WSD

To further understand the performance of our models, we extend our experiment to the original WSD. One example of WSD is presented in Table 6. We experiment our best performing model on SemEval-

Systems	noun	verb	adj	other	SemEval-15
WordNet S1	67.6	50.3	74.3	80.9	67.8
MFS	67.7	49.8	73.1	80.5	67.1
HCAN	72.7	58.2	77.4	84.1	72.8
GlossBERT	79.8	67.1	79.6	87.4	80.4
BEM	81.4	68.5	83.0	87.9	81.7
SimCSE-Unsup	71.2	53.0	87.5	82.5	70.2

Table 7: Performance on SemEval-15 dataset with the original WSD task. MFS denotes the most frequent senses of each word in SemCor [25].

15 [23]. We compare our model with common WSD baselines including the first sense in the WordNet (denote as WordNet S1), Most frequent senses of each word in SemCor [25], HCAN [45](which includes definitions in a neural WSD classifier), GlossBERT [46] (which constructs context- definition pairs for fine-tuning), and BEM [9]. Notice that HCAN, GlossBERT, and BEM are supervised models trained on 37,176 sentences with 802,443 target tokens, while ours can be considered as an unsupervised approach. The results are presented in Table 7, from which we can observe that our method performs reasonably on the data and outperforms the WordNet S1 and MFS baselines, yet other supervised baselines still perform better than ours. One reason behind this can be that senses from WordNet are extremely fine-grained [47]. Without extensive fine-tuning, it is hard for our model to distinguish the detailed differences among senses of a single word. In the future, we will explore on how models perform with merged senses.

7 Conclusion

In this project, we present a novel benchmark, Word Sense Matching, for evaluating the PTLMs' capability in understanding definitions. WSM is created from context-definition pairs from various sense inventories with both local and global negative sampling, which forms a solid base to validate different models in their ability to align words with their meanings. The relief from target-centered multiple-choice format of the original WSD to binary classification also potentially en-powered simple downstream application.

We compare various strategies for solving the task and find the performance gain for PTLMs comes from scoring the contextual similarity between contexts and definitions, which reveals potential challenges for the PTLMs to understand human languages: distinguishing senses from words and their antonyms and understanding abstract senses.

In the future, we plan to first explore more complicated architectures for applying PTLMs than the current methods. Then, we explore forming the dataset as a ranking or generation task to test the characteristics of alignment acquired from different objectives. Finally, we seek to test on potential downstream application with our proposed task formation, including text simplification with human annotation, conceptualization (e.g., detect how likely a "dog" in a sentence can be replaced by a "mammal"), and novel object detection with video captioning models.

8 Acknowledgements

We would like to thank our project mentor, Prof. Chris Manning, for his insightful advice and comprehensive knowledge over various areas in natural language processing regarding our proposal, milestone, and potential directions. We are grateful to all instructors, coordinators, and teaching assistants for preparing and presenting a wonderful course with encouraging lectures, well-guided assignments, and the challenging but illuminating final project in this great quarter despite all the turbulence.

References

[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT 2019*, pages 4171–4186, 2019.

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

[3] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. 2018.

[4] Michele Bevilacqua, Tommaso Pasini, Alessandro Raganato, and Roberto Navigli. Recent trends in word sense disambiguation: A survey. In Zhi-Hua Zhou, editor, *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 4330–4338. International Joint Conferences on Artificial Intelligence Organization, 8 2021. Survey Track.

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

[6] Sawan Kumar, Sharmistha Jat, Karan Saxena, and Partha Talukdar. 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, July 2019. Association for Computational Linguistics.

[7] Luyao Huang, Chi Sun, Xipeng Qiu, and Xuanjing Huang. Glossbert: BERT for word sense disambiguation with gloss knowledge. *CoRR*, abs/1908.07245, 2019.

[8] Michele Bevilacqua and Roberto Navigli. Breaking through the 80% glass ceiling: Raising the state of the art in word sense disambiguation by incorporating knowledge graph information. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2854–2864, Online, July 2020. Association for Computational Linguistics.

[9] Terra Blevins and Luke Zettlemoyer. Moving down the long tail of word sense disambiguation with gloss informed bi-encoders. In *Proceedings of the 58th Association for Computational Linguistics*, 2020.

[10] William A. Gale, Kenneth Ward Church, and David Yarowsky. One sense per discourse. In *HLT*, 1992.

[11] Eric Huang, Richard Socher, Christopher Manning, and Andrew Ng. Improving word representations via global context and multiple word prototypes. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 873–882, Jeju Island, Korea, July 2012. Association for Computational Linguistics.

[12] Jiwei Li and Dan Jurafsky. Do multi-sense embeddings improve natural language understanding? In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1722–1732, Lisbon, Portugal, September 2015. Association for Computational Linguistics.

[13] Mohammad Taher Pilehvar and José Camacho-Collados. Wic: 10, 000 example pairs for evaluating context-sensitive representations. *CoRR*, abs/1808.09121, 2018.

[14] Tomás Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. *CoRR*, abs/1310.4546, 2013.

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

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

[17] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020.

[18] Luke Vilnis and Andrew McCallum. Word representations via gaussian embedding. *International Conference on Learning Representations*, 12 2015.

[19] Luke Vilnis, Xiang Li, Shikhar Murty, and Andrew McCallum. Probabilistic embedding of knowledge graphs with box lattice measures. *arXiv preprint arXiv:1805.06627*, 2018.

[20] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*, 2019.

[21] Tianyu Gao, Xingcheng Yao, and Danqi Chen. SimCSE: Simple contrastive learning of sentence embeddings. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2021.

[22] Martin Joos. Machine translation of languages: Fourteen essays, 1956.

[23] Andrea Moro and Roberto Navigli. 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, June 2015. Association for Computational Linguistics.

[24] Kaveh Taghipour and Hwee Tou Ng. One million sense-tagged instances for word sense disambiguation and induction. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning*, pages 338–344, Beijing, China, July 2015. Association for Computational Linguistics.

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

[26] Rui Suzuki, Kanako Komiya, Masayuki Asahara, Minoru Sasaki, and Hiroyuki Shinnou. All-words word sense disambiguation using concept embeddings. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, May 2018. European Language Resources Association (ELRA).

[27] Yoav Levine, Barak Lenz, Or Dagan, Ori Ram, Dan Padnos, Or Sharir, Shai Shalev-Shwartz, Amnon Shashua, and Yoav Shoham. SenseBERT: Driving some sense into BERT. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4656–4667, Online, July 2020. Association for Computational Linguistics.

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

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

[30] Michael Lesk. 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, page 24–26, New York, NY, USA, 1986. Association for Computing Machinery.

[31] Florentin Vasilescu, Philippe Langlais, and Guy Lapalme. Evaluating variants of the lesk approach for disambiguating words. In *LREC*, 2004.

[32] Ming Wang and Yinglin Wang. Word sense disambiguation: Towards interactive context exploitation from both word and sense perspectives. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5218–5229, Online, August 2021. Association for Computational Linguistics.

[33] Robert L. Logan IV, Ivana Balažević, Eric Wallace, Fabio Petroni, Sameer Singh, and Sebastian Riedel. Cutting down on prompts and parameters: Simple few-shot learning with language models, 2021.

[34] Tianyu Gao, Adam Fisch, and Danqi Chen. Making pre-trained language models better few-shot learners. In *Association for Computational Linguistics (ACL)*, 2021.

[35] Derek Tam, Rakesh R Menon, Mohit Bansal, Shashank Srivastava, and Colin Raffel. Improving and simplifying pattern exploiting training. 2021.

[36] Guanghui Qin and Jason Eisner. Learning how to ask: Querying LMs with mixtures of soft prompts. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5203–5212, Online, June 2021. Association for Computational Linguistics.

[37] Elad Ben Zaken, Shauli Ravfogel, and Yoav Goldberg. Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. *CoRR*, abs/2106.10199, 2021.

[38] Leandra Fichtel, Jan-Christoph Kalo, and Wolf-Tilo Balke. Prompt tuning or fine-tuning - investigating relational knowledge in pre-trained language models. In *3rd Conference on Automated Knowledge Base Construction*, 2021.

[39] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA, October 2013. Association for Computational Linguistics.

[40] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020.

[41] Timo Schick and Hinrich Schütze. Exploiting cloze-questions for few-shot text classification and natural language inference. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 255–269, Online, April 2021. Association for Computational Linguistics.

[42] Ethan Perez, Douwe Kiela, and Kyunghyun Cho. True few-shot learning with language models. *NeurIPS*, 2021.

[43] Ben Wang and Aran Komatsuzaki. GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. `https://github.com/kingoflolz/mesh-transformer-jax`, May 2021.

[44] Xin Li and Dan Roth. Learning question classifiers. In *COLING 2002: The 19th International Conference on Computational Linguistics*, 2002.

[45] Fuli Luo, Tianyu Liu, Qiaolin Xia, Baobao Chang, and Zhifang Sui. Incorporating glosses into neural word sense disambiguation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2473–2482, Melbourne, Australia, July 2018. Association for Computational Linguistics.

[46] Luyao Huang, Chi Sun, Xipeng Qiu, and Xuanjing Huang. 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, November 2019. Association for Computational Linguistics.

[47] Rion Snow, Sushant Prakash, Daniel Jurafsky, and Andrew Y. Ng. Learning to merge word senses. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 1005–1014, Prague, Czech Republic, June 2007. Association for Computational Linguistics.