Can Pre-trained Language Models Understand Definitions?

Tina Li Xiaoyuan Ni Xinran Zhao

Department of Computer Science, Stanford

Introduction

Humans are great in giving definition on the concepts they meet. What about Pre-trained Language Models? Some previous studies and ours are in Figure 1.

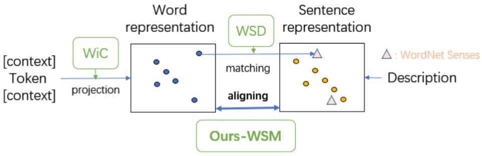

Figure 1. WiC, WSD, and our WSM, three tasks focus on different stages of understanding descriptive sentences.

Word sense disambiguation (WSD) is commonly formed as a multiple choices task that requires models to choose the correct sense of a word in its context from candidates in the sense inventory, such as WordNet [1].

WiC is formed as a binary classification task that requires models to decide if a target word in two sentences share a similar meaning.

We propose Word Sense Matching (WSM) as a binary classification task focusing on finding the alignment of representation space of words in context and their possible descriptions.

Interesting characteristics of WSM:

- the descriptive sentences are collected from multiple sources and not limited by a single sense inventory;
- as a binary classification task with three factors: word, context, description, WSM suitable for evaluating various objectives and methods.

Background: Representation Learning

Learning the representation of languages has been an important task in NLP.

- **Word Embedding:** from One-hot to Glove, CBOW, to contextualized BERT, RoBERTa, and etc.
- **Entity Embedding:** from averaged word embeddings to Gaussian Embedding, Box Embedding.
- **Sentence Embedding** from averaged word embeddings to contrastive SBERT, SimCSE.

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 descriptive sentences.

Background: Word Sense Disambiguation

The origin of the task of word sense disambiguation (WSD) can be sourced from *Machine translation of languages: Fourteen essays* in 1956, where the use of word senses in machine translation is recognized. The task gained popularity from the community through the years.

- **Datasets:** Senseval-2,3; SemEval-07,13,15; SemCor; OMSTI.
- **Methods:** Lesk, HCAN, EWISE, GlossBERT, BEM, EWISER.

Comparing to WSD, we relieve the reliance on explicit sense labels from the inventories and ask the models to generally decide if a definition is correct, which allows more space for application.

Dataset Creation

Split	Instance	Avg. Context Length	Avg. Definition Length	# Adjective	# Noun	# Verbs
Training	6,240	5.91	7.78	2,698	1,584	1364
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.	Someone that causes trouble.
0	perceivable	It is perceivable through the mist.	The act of looking for something.

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

To further capture the dynamic nature of word semantics, we follow two guidelines to create our WSM dataset: (1) multiple sources: previous human annotations (i.e., SemCor), sense inventories such as WordNet[1], and commonly used dictionaries (e.g., Oxford, Webster, and Cambridge dictionaries); (2) multiple sampling: conducting negative sampling from both senses of the target words and senses of other words.

Human evaluation: 3 college students on 50 questions: 92.67% inter-annotator agreement (IAA), *good agreement!* Statistics and examples are given in Table 1, 2, respectively.

Methodology: *Disjoint vs. joint*

Whether we encode the context and the definition separately or jointly? Different formulation leads to different time complexity during application: Suppose that we 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.

1. ***Disjoint:*** Following the widely accepted bi-encoder framework in WSD, we separately encode the context and definitions and learn the distance metric of the encoded vectors to acquire the prediction.
2. ***Joint:*** Following the recent trend in utilize pre-trained language models with prompt-based tuning, we synthesize main factors (word, context, and definition) into a single sentence and make prediction over its embedding. One possible way to generate the prompt is "In <*context*>, the <*target*> can be described as <*definition*>"

Methodology: *Prompt Design*

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 design.

Designing good prompt pattern is crucial to the success of prompt-based tuning.

We explore four kinds of patterns: (1) manual: we manually create a sentence to link the factors (i.e., contexts, targets, and definitions); (2) trigger: we add a few randomly initialized soft tokens between the factors; (3) sep: we add a <sep> token between each of the factors; (4) null: we add a blank between each of the factors. Besides fine-tuning all parameters, we also explore training the bias terms only as previous work (Bitfit). Pioneer results are presented in Table 3.

Quick solution: For WSM, use all parameters and null prompts.

Methodology: *Zero-, Few-, and Many-shot*

The recent progress in PTLMs has led to the advance of learning a task with only a few examples (i.e., Few-shot Learning). This setting is important since it is possible that the data is not rich when we try to apply WSM models in new domains (e.g., medical domain).

1. ***Zero-shot methods:*** Lesk, GPT-J, unsupervised SimCSE.
2. ***Few-shot and many-shot method:*** both ***disjoint*** and ***joint*** settings with various pre-trained language models: bert-base, bert-large, robert-base, roberta-large.

Results and Analysis

Method	Source						Overall
	WN (400)	Collins (56)	Longman (90)	Webster (78)	Oxford (90)	Cambridge (66)	(780)
				Zero-shot			
Lesk	55.5%	71.43%	73.33%	67.95%	74.44%	72.73%	63.59%
GPT-J	49.75%	57.14%	61.11%	62.82%	61.11%	56.06%	54.74%
SimCSE-unsup	77.75%	80.36%	90.00%	85.90%	74.44%	80.30%	**80.00%**
				Few-shot			
joint - BB	54.75%	55.36%	57.78%	57.69%	51.11%	53.03%	54.87%
joint- BL	52.75%	62.50%	67.78%	71.79%	61.11%	65.15%	59.10%
joint- RB	52.50%	50.00%	52.22%	56.41%	47.78%	54.55%	52.31%
joint- RL	56.00%	55.36%	55.56%	51.28%	44.44%	51.52%	53.72%
disjoint- BB	71.25%	69.64%	76.67%	82.05%	67.78%	66.67%	72.05%
disjoint- BL	71.75%	82.14%	90.00%	79.49%	76.67%	74.24%	76.15%
disjoint- RB	75.25%	80.36%	90.00%	85.90%	70.00%	69.70%	77.31%
disjoint- RL	74.75%	69.64%	85.56%	83.33%	73.33%	77.27%	76.54%
				Full Fine-tuning			
joint - BB	73.25%	73.21%	75.56%	82.05%	74.44%	74.24%	74.62%
joint- BL	75.25%	73.21%	83.33%	79.49%	64.44%	72.73%	75.00%
joint- RB	69.50%	76.79%	73.33%	73.08%	68.89%	72.73%	71.03%
joint- RL	70.50%	73.21%	72.22%	66.67%	71.11%	71.21%	70.64%
disjoint- BB	73.25%	73.21%	75.56%	82.05%	74.44%	74.24%	74.62%
disjoint- BL	71.75%	78.57%	86.67%	83.33%	73.33%	74.24%	75.51%
disjoint- RB	75.50%	78.57%	90.00%	85.90%	68.89%	69.70%	77.18%
disjoint- RL	75.00%	67.86%	85.56%	84.62%	72.22%	75.76%	76.41%

Table 4. The statistics and performances of different methods over our test set.

1. ***Zero-shot learning works well.*** Unsupervised SimCS model see 1 million sentences.
2. ***Joint models only work after fine-tuning.*** Resulted prompt can be too hard to be solved directly by inductive bias of pre-trained language models.
3. ***Fine-tuning is not necessary for disjoint models.*** Good sentence embedding models already contain enough information.
4. ***Performance differs on data from different sources.*** It is important to include multiple sense inventories in the future.

References

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

This page is forked from rylanschaeffer.github.io

xzhaoar@stanford.edu