Multi-fusion on SemEval-2013 Task13: Word Sense Induction

Hui-Qiang Jiang **Da-Wei Lee**
Peking University
{1801210840,1701210963}@pku.edu.cn

Abstract

In 2019, the word representing technique, the embedding, is benefitting all the NLP field. Since it's much better than in 2013. Even our basic Top N model beats the best model in 2013 almost twice on Fuzzy NMI and over 8% on Fuzzy B-Cube. And the BiLM with clustering model has even more significant improvement on the result. As we can see how we can transform the meaning of a word which is ambiguity into a computable format is a really important task in NLP.

1 Introduction

Word Sense Induction(WSI) is a typical Natural Language Processing task. It has given some ambiguous word and some sample texts which may have multi-meaning And the task is to discover the multiple sense or meanings. The problem of WSI has been extensively studied with a series of shared task on the topic like *SemEval* 2013 Task13. Previous works on WSI used context vectors and LDA, BiLM.

SemEval-2013 Task 13 [4] has three subtasks (but the description link of each subtask has lost)

1. Non-graded Word Sense Induction Subtask
2. Graded Word Sense Induction Subtask
3. Lexical Substitution SubTask

In this paper, we will focus on the Graded WSI Subtask.

Graded WSI

Graded WSI means that for each word sense we will assign a weight for it to represent how much it "means" that sense.

For an example of a gold key: add.v add.v.13 add%2:32:01::/4 add%2:30:00::/2

- Lemma: add
- POS: verb
- Instance: 13
- Meanings
  1. add%2:32:01::/4
     - Definition: state or say further
     - Weight: 4
  2. add%2:30:00::/2
     - Definition: make an addition (to); join or combine or unite with others; increase the quality, quantity, size or scope of
     - Weight: 2

In this subtask, we will attempt in two different ways. First is clustering between the sense of WordNet and the instances, just like the gold key shown above. Another one is clustering between instances themselves.

Evaluation Metrics

There were five evaluation metrics given by the host. But because we were focused on the WSI

subtask, so we will only try to improve the WSI metrics, that is Fuzzy NMI, Fuzzy B-Cubed and their geometric mean.

2 Approach

2.1 TopN

The naive solution to this competition is to predict/induce one or a few word sense for each instance of all cases. The thought is simply that for each sentence of an instance, we try to calculate its similarity of all the meaning in WordNet 3.1 by transforming them into sentence embedding.

There are some key part of this approach

- The choice of the embedding
- How to construct the sentence embedding
- Determine the similarity between the senses
- Selecting the top N results

And here is the overall procedure of the TopN approach

1. Use the definitions of the word from WordNet (e.g. get the meanings of ADD in VERB)
2. Transfer sentences into Vector (Dataset Instances and WordNet definitions)
3. Calculate the similarity of each definition sentences with the word
4. For each test instance export top N possible sense and using the similarities as weights

Embedding

We've tried to use the fastText pre-trained embedding [5], it's a 300 dimension embedding trained on Wikipedia 2017, UMBC webbase corpus and statmt.org news dataset.

And we've also tried to use BERT [3] pre-trained model to generate embedding. Because we were using the BERT-Base model, the output dimension of embedding is 768.

Sentence Embedding

To construct the word embedding into sentence embedding, we've tried the following methods.

- Naive Adding
- Naive Adding with Normalization with sentence length
- Padding the sentence to max sentence length with the "average embedding"

Among them, the second one which normalized the summation of the embeddings has the best result.

Similarity

To determine the similarity between the WordNet synset definition between the instances of the dataset.

We've tried the following distance calculation: (we use its inverse as similarity)

- Cosine
- Euclidean
- Minkowski

And found that cosine will generate the best result. Thus in the rest of the experiment, we'll use cosine similarity.

We've also applied a trick on the TopN result. For an example of the top similarities is [7, 6, 6.5, 5, 4] and we want to get the Top 3 result. Initially, we'll get [7, 6, 6.5]. But we'll minus the Top N+1's value that makes the result become [2, 1, 0.5], which greater the ratio (importance) between them. And we get the better result after this.

Strict and Generalized Top N

Sometimes a word could have only one sense or multiple sense, so we've tried to predict the sense with a threshold that the results amount will differ.

So the Generalized Top N model means the maximum prediction will be N but can be less than N but at least one.

But in the end, we found that for each generalized version Top N will be about 0.X less than the strict one. We were guessing this is because the prediction order of the similarities was already wrong. So when we strip the result with a threshold. We may get rid of the correct one but with lower weight. This will lower out prediction.

And finally, we found that, when N = 2, we'll get the best result.

2.2 BiLM

In front of the paper, we propose a Word Sense Induction method base on WordNet and pretrain Language Model. But for more thought, we only extract the paragraph grained representations though it's so similarity to the meaning of the center word. And we only match the word meaning from WordNet. It has some deviation between actual meaning and WordNet meaning. For the more, In our method, the rule of division top-N is the decision by practical experience. It somethings don't work well.

For a naive mind, we can change the paragraph grained to the word grained and then clustering word meaning from sample sentences. It means that we should build a more fine-grained from around word rather than common word embedding, like ELMo [7], Bert. But if we directly replace word to context word embedding, It's too much noise in the word embedding spacing between same meaning word. It causes that direct cluster ELMo embedding vector don't better than other pretrain word embedding. Refer the Bask's Work [2]. We using substitute vectors instead of directing using word embedding producing by ELMo. It's mean that we choose some word which is the similarity to the meaning of this word in their sentence. The way of choose substitute word [2] is base on reusing Language Model ELMo. In addition to that work, we also use a skill to get unequal length similarity meaning substitute word. The most similarity substitute word will be cluster to several clusters.

Substitute Vectors Language Model is an intuitive way we to get the meaning of the word from context word from sample sentence. But for the most time, directly replace the word by word embedding is meaning amplification cluster distance. It has low robustness when the word vector causing by Language Model have noise. So refer to the work of Neural biLM [1], we use BiLM to change word vector to a similar word. This way reduce the word noise meaning, and amplification the important meaning. And in the actual, We random choose most similarity word base Language Model weight dot local word vector. By multi choosing, we get the most similarity word no matter what the num of word most similarity.

BiLM & Cluster embedding Language Model is an obvious idea in Word Sense Induction problem. In the Substitute Word process, we using Language Model for twice. First, we extract context word meaning to center word vector by forwarding Language Model. And we random choose the most similarity word from the Matrix of Language model weight dot center word embedding. It's product by backward Language Model. It's a process of word vector to the word. And then we use the substitute word to the cluster, It also needs a word embedding process though It does not have context word. So In our second method, we multi-use Language Model to extract words actual meaning.

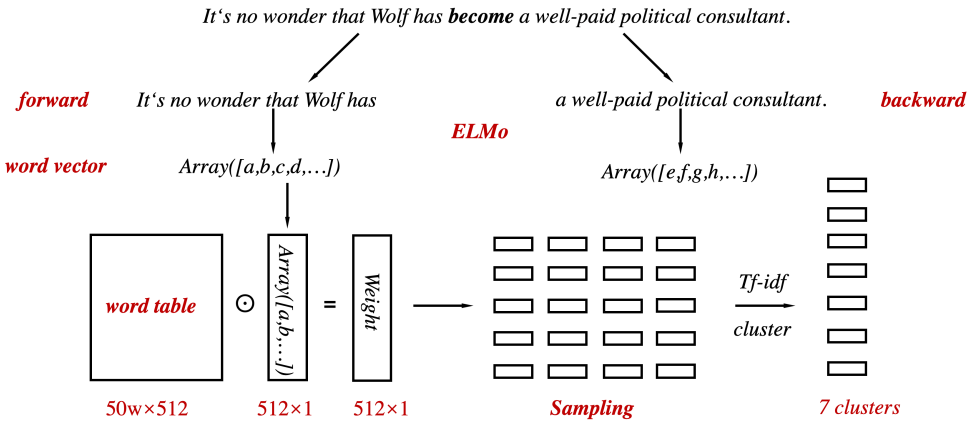

Figure 1: The overview of BiLM & cluster model structure

3 Experiment

TopN Result

Strict TopN Approach with Naive Adding

In this experiment, we will use the most naive way to construct sentence embedding but try to see which N and which similarity is better. (Table 1)

Experiment		WSI Metrics		
Model	Similarity	Fuzzy NMI	Fuzzy B-Cubed	Average
Top 1	Cosine	2.81	50.22	11.89
Top 2	Cosine	8.56	52.10	21.12
Top 3	Cosine	7.15	42.87	17.51
Top 4	Cosine	6.61	32.07	14.56
Top 5	Cosine	5.92	24.80	12.11
Top 1	Euclidean	3.63	47.78	13.17
Top 2	Euclidean	7.87	46.06	19.04
Top 3	Euclidean	6.97	40.36	16.77
Top 4	Euclidean	6.46	34.25	14.87
Top 5	Euclidean	6.15	28.33	13.20
Top 1	Minkowski	3.63	47.78	13.17
Top 2	Minkowski	7.87	46.06	19.04
Top 3	Minkowski	6.97	40.36	16.77
Top 4	Minkowski	6.46	34.25	14.87
Top 5	Minkowski	6.15	28.33	13.20
Top 2 (BERT)	Cosine	6.74	46.78	17.76

Table 1: Comparison of different Strict Top N model and different similarity with fastText embedding

Different TopN Approach with Different Sentence Embedding

In this experiment, we test the combination of strict or generalized top N and using which sentence embedding. And shows the best result of each setting. (Table 2)

Experiment		WSI Metrics		
Model	Sentence Embedding	Fuzzy NMI	Fuzzy B-Cubed	Average
Strict Top 2	Naive Adding	8.56	52.10	21.12
Generalized Top 2	Naive Adding	8.35	52.40	20.92
Strict Top 2	Normalized Adding	**8.92**	**52.51**	**21.64**
Strict Top 2	Normalized Padding	8.55	46.58	19.95

Table 2: Comparison between different Top N model and different sentence embedding

BiLM Result

BiLM, Substitute Vectors, Cluster

Refer to figure 1, We use BiLM, Substitute Vectors, Cluster to extract multi-sense or meanings word which is ambiguation. In our work, We choose ELMo, FastText, Glove as the BiLM model. The ELMo pre-training model is from allennlp. The fastText pre-training model is from fasttext.cc. The Glove[6] pre-training model is from Stanford.

Before the clustering processing, we need a representatives method to extract substitute word meaning. In this processing, we test for one-hot, TF-IDF, FastText, Glove, Bert, ELMo. we used *sklearn* for both one-hot, TF-IDF weighting and clustering. The Bert model is running in a 12-head multi-transformer model. And the hidden state as the out params.

In backward substitute, we choose the random choose frequent is 4 words a group, and every sample sentence we random sampling for 20 times. And the cluster num is 7. We use agglomerative clustering (cosine distance, average linkage) and induce a fixed number of clusters.

In this experiment, we will try to replace the original ELMo language model and see the effect of using different clustering target. (Table 3)

From the experiment result, We found ELMo + TF-IDF is the best combination in our work. Treating each representative as a document, TF-IDF reduces the weight of uninformative words shared by many representatives.

Test for Clustering Num

And We also test for the cluster num. And we found in this DataSet, 7 clusterings is the best params. For more thought, when we defined the cluster num, It's a similarity to the top-N num. It's decided by the data feature.

Best Result

Following will list the best result among our models and the best model in the 2013 competition. (Table 5)

Experiment		WSI Metrics		
Language Model	Cluster	Fuzzy NMI	Fuzzy B-Cubed	Average
ELMo	one-hot	9.28	58.70	23.34
ELMo	TF-IDF	11.06	57.72	25.27
ELMo	BERT	2.65	54.34	12.00
ELMo	GloVe	8.28	60.90	22.45
ELMo	fastText	7.40	61.39	21.32
GloVe	one-hot	8.66	59.21	22.64
GloVe	TF-IDF	10.69	57.68	24.83
GloVe	GloVe	8.08	60.88	22.18
GloVe	fastText	6.49	61.08	19.91
fastText	one-hot	8.78	58.59	22.68
fastText	TF-IDF	10.82	57.34	24.90
fastText	GloVe	7.79	60.52	21.72
fastText	fastText	7.08	61.11	20.80

Table 3: Comparison using different language model and using different clustering target

ClusterNum	Fuzzy NMI	Fuzzy B-Cubed	Average
5	9.68	**58.88**	23.87
6	10.21	58.25	24.39
7	11.06	57.72	25.27
8	10.88	56.60	24.82
9	**11.58**	56.61	**25.60**
10	10.95	55.94	24.75

Table 4: BiLM in multiple Cluster NUm (in %)

Experiment		WSI Metrics		
Team	System	Fuzzy NMI	Fuzzy B-Cubed	Average
PKU NLP ForFun (Our)	Strict Top 2 (Normalized)	8.92	52.51	21.64
Neural BiLM (2018)	ELMo + TF-IDF	11.06	57.72	25.27
AI-KU (2013)	Base	4.50	35.10	12.57
Unimelb (2013)	50k	3.90	44.10	13.11

Table 5: Comparison between our model and the best model in the competition

4 Conclusion

In our own designed approach, we found that the model "Strict Top 2 with fastText embedding and using Cosine similarity and form the sentence embedding with Naive Adding Normalized with sentence length" will generate the best result.

The BiLM + clustering model has obvious improve than the approach base on WordNet. But our work in BiLM doesn't have obvious improve than the origin author result.

We think that because, in 2013, the embedding technique is not so mature. So we have taken advantage of the embedding, and thus the result beats all the best model in 2013.

References

[1] AMRAMI, A., AND GOLDBERG, Y. Word sense induction with neural biLM and symmetric patterns. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing* (Brussels, Belgium, Oct.-Nov. 2018), Association for Computational Linguistics, pp. 4860–4867.

[2] BASKAYA, O., SERT, E., CIRIK, V., AND YURET, D. Ai-ku: Using substitute vectors and co-occurrence modeling for word sense induction and disambiguation. In *Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)* (2013), vol. 2, pp. 300–306.

[3] DEVLIN, J., CHANG, M.-W., LEE, K., AND TOUTANOVA, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[4] JURGENS, D., AND KLAPAFTIS, I. 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)* (2013), vol. 2, pp. 290–299.

[5] MIKOLOV, T., GRAVE, E., BOJANOWSKI, P., PUHRSCH, C., AND JOULIN, A. Advances in pre-training distributed word representations. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)* (2018).

[6] PENNINGTON, J., SOCHER, R., AND MANNING, C. D. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)* (2014), pp. 1532–1543.

[7] PETERS, M. E., NEUMANN, M., IYYER, M., GARDNER, M., CLARK, C., LEE, K., AND ZETTLEMOYER, L. Deep contextualized word representations. In *Proc. of NAACL* (2018).