Introduction to NLP

Word Sense Disambiguation

Abhishek and Yash

IIIT- Hyderabad

April 25, 2023

Table of Contents I

Introduction

Word Sense Disambiguation (WSD) is a crucial task in natural language processing, aimed at automatically determining the meaning of words in context. It involves associating a word with its most appropriate sense from a pre-defined sense inventory, such as WordNet, which is the de-facto inventory for English in WSD.

Dataset

The SemCor 3.0 dataset is a sense-tagged subset of the Brown Corpus, where each word in the corpus has been labeled with its sense according to WordNet 2.0. It contains around 235,000 words, and has been widely used as a benchmark dataset for word sense disambiguation tasks. SemCor 3.0 is considered to be a challenging dataset due to its size, diversity of texts and the ambiguity of words in context.

	uid	sentence	sense_keys	glosses	target
0	d000.s000.t000	How [TGT] long [TGT] has it been since you rev...	['long%3:00:04::', 'long%3:00:02::', 'long%3:0...	['(of speech sounds or syllables) of relativel...	[0,1]
1	d000.s000.t001	How long has it [TGT] been [TGT] since you rev...	['be%2:42:06::', 'be%2:42:02::', 'be%2:42:13::...	['be identical to; be someone or something', '...	[3]
2	d000.s00g0.t002	How long has it been since you [TGT] reviewed ...	['review%2:31:04::', 'review%2:32:00::', 'revi...	["refresh one's memory", 'appraise critically'...	[2]
3	d000.s000.t003	How long has it been since you reviewed the [T...	['objective%1:06:00::', 'objective%1:09:00::']	['the lens or system of lenses in a telescope ...	[1]
4	d000.s000.t004	How long has it been since you reviewed the ob...	['benefit%1:21:00::', 'benefit%1:10:00::', 'be...	['financial assistance in time of need', 'a pe...	[0]

Figure: Prepared Data

Baseline

We Used three approaches :

- KNN
- Naive Bayess
- Lesk

Accuracy			
Algorithms	brown1	brown2	brown1 + brown2
K-Nearest Neighbour	67.132%	67.532%	71.826%
Naive Bayes (context)	51.002%	51.341%	53.112%
Naive Bayes (context + pos)	40.231%	42.541%	44.402%
Simple Lesk	34.358%	34.859%	34.645%
Extended Lesk	51.717%	52.957%	52.485%

Figure: Baseline Accuracies

Training

The model is trained for 30,000 sentences in the semcor dataset for 10 epochs. For 10 epochs , the losses and accuracy for the training set and validation set are given below:

```
Epoch : 1/10 | Loss : 0.8970 | Accuracy : 0.6899
Epoch : 1/10 | Validation Loss : 0.7616 | Validation Accuracy : 0.7095
Epoch : 2/10 | Loss : 0.7125 | Accuracy : 0.7450
Epoch : 2/10 | Validation Loss : 0.7578 | Validation Accuracy : 0.7108
Epoch : 3/10 | Loss : 0.7048 | Accuracy : 0.7512
Epoch : 3/10 | Validation Loss : 0.7538 | Validation Accuracy : 0.7161
Epoch : 4/10 | Loss : 0.7021 | Accuracy : 0.7534
Epoch : 4/10 | Validation Loss : 0.7518 | Validation Accuracy : 0.7160
Epoch : 5/10 | Loss : 0.7028 | Accuracy : 0.7527
Epoch : 5/10 | Validation Loss : 0.7523 | Validation Accuracy : 0.7150
Epoch : 6/10 | Loss : 0.7012 | Accuracy : 0.7541
Epoch : 6/10 | Validation Loss : 0.7522 | Validation Accuracy : 0.7156
Epoch : 7/10 | Loss : 0.7003 | Accuracy : 0.7546
Epoch : 7/10 | Validation Loss : 0.7550 | Validation Accuracy : 0.7143
Epoch : 8/10 | Loss : 0.6991 | Accuracy : 0.7561
Epoch : 8/10 | Validation Loss : 0.7530 | Validation Accuracy : 0.7135
Epoch : 9/10 | Loss : 0.6992 | Accuracy : 0.7558
Epoch : 9/10 | Validation Loss : 0.7548 | Validation Accuracy : 0.7120
Epoch : 10/10 | Loss : 0.6984 | Accuracy : 0.7566
Epoch : 10/10 | Validation Loss : 0.7549 | Validation Accuracy : 0.7125
```

Figure: Training

Results

Results		
Dataset	Accuracy	Loss
SemEval-2013	64.86%	0.7982
SemEval-2015	57.42%	0.8796
SensEval2	63.54%	0.8299
SensEval3	60.59 %	0.9417
All-merged	61.82%	0.8730

Figure: BiLSTM Results

BERT-WSD

In our work, we are adapting BERT for Word Sense Disambiguation (WSD) using the Gloss Selection Objective and Example Sentences. We are using a similar approach to GlossBERT, where we fine-tune BERT on sequence-pair binary classification tasks. Our training data consists of context-gloss pairs, where each pair contains a sentence with a target word to be disambiguated (context) and a candidate sense definition of the target word (gloss) from a lexical database such as WordNet.

Model

```
BERT_for_WSD(
  (bert): BertModel(
    (embeddings): BertEmbeddings(
      (word_embeddings): Embedding(30522, 768, padding_idx=0)
      (position_embeddings): Embedding(512, 768)
      (token_type_embeddings): Embedding(2, 768)
      (LayerNorm): LayerNorm((768,), eps=1e-12, elementwise_affine=True)
      (dropout): Dropout(p=0.1, inplace=False)
    )
    (encoder): BertEncoder(
      (layer): ModuleList(
        (0-11): 12 x BertLayer(
          (attention): BertAttention(
            (self): BertSelfAttention(
              (query): Linear(in_features=768, out_features=768, bias=True)
              (key): Linear(in_features=768, out_features=768, bias=True)
              (value): Linear(in_features=768, out_features=768, bias=True)
              (dropout): Dropout(p=0.1, inplace=False)
            )
            (output): BertSelfOutput(
              (dense): Linear(in_features=768, out_features=768, bias=True)
              (LayerNorm): LayerNorm((768,), eps=1e-12, elementwise_affine=True)
              (dropout): Dropout(p=0.1, inplace=False)
            )
          )
          (intermediate): BertIntermediate(
            (dense): Linear(in_features=768, out_features=3072, bias=True)
            (intermediate_act_fn): GELUActivation()
          )
          (output): BertOutput(
            (dense): Linear(in_features=3072, out_features=768, bias=True)
            (LayerNorm): LayerNorm((768,), eps=1e-12, elementwise_affine=True)
            (dropout): Dropout(p=0.1, inplace=False)
          )
        )
      )
    )
    (pooler): BertPooler(
      (dense): Linear(in_features=768, out_features=768, bias=True)
      (activation): Tanh()
    )
  )
  (dropout): Dropout(p=0.1, inplace=False)
  (ranking_linear): Linear(in_features=768, out_features=1, bias=True)
)
```

Result

We trained our BERT-based model on a dataset of 10,000 records from SemCor 3.0, for a total of 3 epochs. This was due to constraints on both computational resources and time. We then tested our model on multiple datasets, including SemEval-2013, SemEval-2015, SensEval2, SensEval3 as well as a combination of all three.

Results	
Dataset	Accuracy
SemEval-2013	74.8783%
SemEval-2015	78.767%
SensEval2	75.021%
SensEval3	70.054 %
All-merged	73.541%

Figure: Test Results

Observation

In non neural models, we used LESK algorithm which is a unsupervised algorithm, whose accuracy is not so good whereas for KNN the accuracy is little high than Naive Bayes (both are supervised). BERT-WSD outperformed all the models and is doing good.
The transformer-based architecture used in BERT, along with its self-supervised pre- training, attention mechanism, and bidirectional modeling, allows it to capture more complex relationships between words and their contexts, making it a highly effective model for tasks like word sense disambiguation.

References

1. *Adapting BERT for Word Sense Disambiguation with Gloss Selection Objective and Example Sentences -Boon Peng Yap, Andrew Koh, Eng Siong Chng*
2. *Rezapour, A., Fakhrahmad, S.M., Sadreddini, M.H. Applying Weighted KNN to Word Sense Disambiguation*
3. *High WSD accuracy using Naive Bayesian classifier with rich features -Cuong Anh Le and Akira Shimazu*