Applying Neural Networks in Word Sense Disambiguation: A literature review

Zhuoru Lin

New York University zlin@nyu.edu

Abstract

Word sense disambiguation(WSD) has long been an open computational linguistic task. Historically, assorted WSD algorithms had been proposed. In this review study, the tasks and challenges of WSD will be introduced. Supervised Learning and the most recent semi-supervised Learning algorithm will be discussed. The performances of several Neural Network models will be summarized. Some future improvement suggestions will be made.

1 Introduction

One of the key ambiguity of human languages is the multiplicity of senses associated with one word in different contexts. Take an English word, 'qualify', for example. According to WordNet, seven possible senses (sorted by frequency) are related to the word 'qualify':

- S: (v) qualify, measure up (prove capable or fit; meet requirements)
- S: (v) qualify (pronounce fit or able) "She was qualified to run the marathon"; "They nurses were qualified to administer the injections"
- S: (v) qualify (make more specific) "qualify these remarks"
- S: (v) qualify, dispose (make fit or prepared) "Your education qualifies you for this job"
- S: (v) stipulate, qualify, condition, specify (specify as a condition or requirement in a contract or agreement; make an express demand or provision in an agreement) "The will stipulates that she can live in the house for the rest of her life"; "The contract stipulates the dates of the payments"
- S: (v) qualify, characterize, characterise (describe or portray the character or the qualities or peculiarities of) "You can characterize his behavior as that of an egotist"; "This poem can be characterized as a lament for a dead lover"
- S: (v) modify, qualify (add a modifier to a constituent)

When a word does not exist with its most common sense, ambiguity imposes difficulties for language understanding even for some native speakers. Senses ambiguity also brought challenges to computational linguistics task such as Machine Translation. Word Sense Disambiguation(WSD) is a Natural Language Processing(NLP) task that aims to correctly assign senses to words based on their context. Generally, there exists two type of WSD tasks. In All-words WSD, one model is trained to assign senses to all words in a context. In Lexicon-sample WSD, different models are trained to assign senses respective preselected lexicon samples.

WSD is challenging for numerous reasons. First, the amount of knowledge required by WSD is enormous. For each sense, a model, even a highly efficient one, would need multiple related contextual information in order to extract regularity. Unfortunately, the deficiency and costliness of labeled data is the core weakness of WSD study [Gale et al. 1992]. The evolutionary nature of language aggravates this problem since all corpus must be manually reannotated when senses changes. Second, the human partition of senses is not perfect essentially, and some senses are

partially overlapped in some contexts [Kilgariff, 1998]. The inner-annotator agreements of manual senses annotation for large corpus are generally lower than 85%, showing the vagueness of senses partition [Ng et al, 1999].

In the late 1940s, WSD was already been recognized as a crucial NLP task [Weaver 1949]. But it is not until the 1980s when machine readable senses knowledge became available that researchers made significant progress. With revolution of machine learning swept through statistics, knowledge based, supervised learning, and semi-supervised learning model had been developed for WSD. With current accuracy, WSD is believed to potentially improves Machine Translations [Chen and Ng, 2007].

In this literature review, I will first examine the knowledge sources for a successful WSD model. Then I will briefly introduce some examples of supervised learning and semi-supervised learning neural networks models developed for WSD. Finally, I will compare the performances of different models in both Lexicon Sample WSD and All-words WSD. Finally I will make suggestions of improvement on current models and discuss potential future work. For a thorough survey of WSD, I refer the interested reader to [Navigli, 2009].

2 Knowledge sources for WSD

2.1 Word Sense Inventories

A machine-readable word senses inventory is the fundamental of WSD research. One of the first sense inventories for NLP is the Longman Dictionary of Contemporary English [Proctor,1979]. Currently the most widely used senses inventory for NLP is WordNet [Miller 1995; Fellbaum 1998] which contains senses knowledge of 155,000 words. WordNet also contains synnets which show syntactic relation information between words that are the essential to knowledge based WSD algorithm. The fine-grained word senses of WordNet are discovered to be not ideal for WSD. Recently investigations [Navigli et al, 2007; Yuan et al, 2016] had been made to resort to course-grained word senses inventory such as Oxford Dictionary of English (ODE) and New Oxford American Dictionary(NOAD) [Stevenson, 2010; Stevenson and Lindberg, 2010].

2.2 Annotated Corpus

Annotated corpus enables supervised learning for WSD. SemCor [Miller et al. 1993], with 234,000 WordNet annotated words, is currently the most widely used annotated corpus for All-words WSD. Semantic English Language Database (SELD) is a collection of lexicographer-curated example sentences provided by Oxford University Press. A recent Google research also manually annotated SemCor corpus with NOAD senses and reported to achieve better accuracy for All-word WSD [Yuen et al, 2016]. Senseval (now Semeval) [Kilgariff, 1998], a WSD competition since 1998, also offers annotated corpus for both Lexicons sample WSD and All-words WSD. Semeval corpus usually serves as a test set for objective comparison of different WSD models.

2.3 Word Embeddings

Recently the Artificial Neural Networks, with state-of-art accuracy, attract attentions in WSD study. Neural Networks for NLP leverages the advantage of using word embeddings to represent text syntactic relations. Deploying word embeddings in Neural Networks had proved to significantly improve the accuracy of All-word WSD [Iacobacci et al, 2016]. Among the most popular embeddings are Word2Vec [Mikolov, 2013] and GloVe [Pennington, 2014].

3 Supervised Learning using Neural Network Models in WSD

3.1 Long Short Term Memory

Long Short Term Memory (LSTM) is a form of Recurrent Neural Network (RNN) that can actively memorize and forget previous hidden states. This feature had been naturally adopted to represent word order information in several NLP tasks. For WSD, LSTM was also proposed. Here we discuss two most recent proposed LSTM models in 2016.

Lexicon Sample WSD A bidirectional LSTM is proposed for Lexicon-sample WSD in [Kageback and Salomonsson, 2016]. A pretrained GloVe word embeddings were used to represent context window. Let x_i be the word embedding of i^{th} word in a document D with dimension of $|D|$ and x_n be the word to be disambiguate with S different senses. Two separate LSTMs iterate $x_0, x_1 ... x_{n-1}$ and $x_{|D|}, x_{|D|-1} ..., x_{n+1}$ in opposite directions. Then the concatenated final hidden state is being transformed to S dimension. Softmax is finally applied to classify the most probable sense. The structure of bidirectional LSTM is shown in Figure 1. The model detail tuning parameters and its source codes written in TensorFLow framework are also made public.

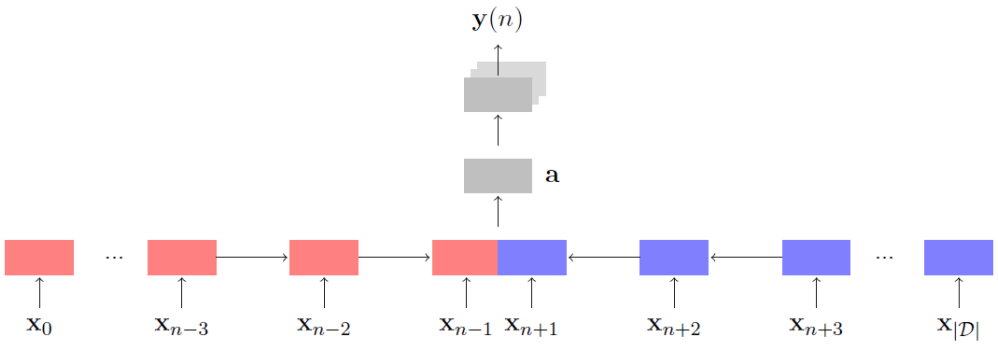

Figure 1: Bidirectional LSTM for Lexicon sample WSD proposed by Kageback and Salomonsson. Figure reprinted from [Kageback and Salomonsson, 2016]

Although reaching the state of art F1 score on Semeval 3, this model should be further scrutinized. First Kageback and Salomonson did not provide enough justification on why using two separate LSTMs instead of one to capture word order information. Adding one more LSTM could potentially increase unnecessary variables to train given the scarce supply of labeled data. This model also can't generalized to solve All-word WSD problem since the dimension of senses classes must be predefined.

All-Words WSD A LSTM supervised learning model is proposed for All-word WSD in [Yuan et al, 2016]. Contrast to Bidirectional LSTM, instead of constructing a classifier predicting probability of sense class, LSTM proposed by Yuan et al predicts the hold out word in a sentence. Let $W_1, W_2, ...$ be words in a context window and $H_1, H_2, ...$ be the hidden state of each corresponding steps in LSTM. First the word to be disambiguate (W_3 in the Figure 2) is replaced with a \$ symbol and EOS is used to specify the end of a sentence. After the LSTM iterate through the entire context window, the final hidden state is first transformed to a context layer which gives the representation of the entire context. Then softmax is applied to to predict the hold out word in the sentence.

To classify senses, the context layer is compared with the sense embeddings of the selected word and the sense with embeddings that has most cosine similarity with the context layer is returned. Each sense embedding is calculated by averaging all contexts embeddings in which target word is labeled with that sense. The structure of this LSTM model is shown in Figure 2.

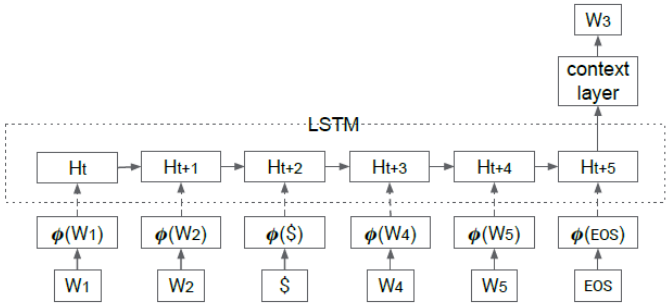

Figure 2: LSTM model proposed by Yuan et al. Figure reprinted from [Yuan et al, 2016]

4 Label Propagation and Semi-Supervised Learning in WSD

To address the problem of lacking labeled corpus, Label Propagation(LP) algorithms were designed so that supervised model in WSD can leverage large unlabeled corpus to make better predictions without over-fitting. A graph-based random walk algorithm, Modified Adsorption (MAD), was proposed in [Talukdar and Crammer, 2009]. Talkdar and Crammer used a graph to represent sense labels of the target words and their relations. Let $G = (V, E, W)$ be an undirected graph. Each vertex $v \in V$ represents a context of the target word. Each edge $e \in E$ represents the similarity between two contexts and weight $w \in W$ represents the strength of this similarity. Suppose each vertex has m possible unique labels. Use $Y_v \in \mathbb{R}^{m+1}$ to represent the prior belief of assigning each label to vertex v. The last dimension of Y_v represents the ignorance of assigning. If equal 1, a predefined dummy label will be assigned to this vertex. Similarly, define $\hat{Y}_v \in \mathbb{R}^{m+1}$ to be the posterior of labels after propagation. MAD strives to achieve three adjectives:

1. For prelabeled v, $Y_v \approx \hat{Y}_v$. This ensures the propagation maintain labels preassigned.
2. For vertex pair v and w, $\hat{Y_v} \approx \hat{Y_w}$ if W_{vw} is large. This ensures closed vertices have similar labels.
3. The discrepancy between $\hat{Y_v}$ and the prior setting of Y_v.

MAD was prove to converge. For detailed implementation algorithm please refer to original paper.

The major drawback of MAD is that it does not scale to large data set since computation is costly when graph becomes large. To address this issue, another random walk algorithm, named Dist-Expander was proposed in [Ravi and Diao, 2015] and was used by Yuan et al in [Yuan et al, 2016]. Dist-Expander was also designed to be optimized for distributed computation.

5 Results and Comparison

For a fair comparison of model, we examine models' performance in Senseval and Semeval competition.

Accuracy and F1-Score Different literature have been using different metrics for evaluation. For Senseval and Semeval accuracy and F1 score are two metrics frequently used. Accuracy is defined by:

$$Accuracy(R) = \frac{\text{Number of correct senses assigned}}{\text{Number of words in total}}. \quad (1)$$

Note that The total number of senses assigned could be smaller than the number of words in total. The accuracy defined above is sometimes referred as 'Recall' in some literature (Thus he). The fraction of correct senses assignment and total senses assigned is defined to be the precision(P) of

the model.

$$Precision(P) = \frac{\text{Number of correct senses assigned}}{\text{Number of senses assigned}}. \quad (2)$$

The model F1-score is defined as:

$$F1 = \frac{2PR}{P+R}. \quad (3)$$

Baseline models Another important comparison is between neural networks models and baseline models. We consider two baseline models:

- Most frequent senses model(MFS): This model always assigns the most frequent sense to every word in test corpus. Note that senses frequencies information is encoded in most sense inventories since modern dictionaries mostly enumerate sense in sense frequency order.
- ExtLesk: This model predicts by comparing the Word2Vec gloss with word context window in a bag-of-word fashion. Model result was provided in [Navigli and Lapata,2010]

Non-Neural-Networks models I would also like to compare neural networks models with other classifiers. Due to the length constraint and the focus of this paper, I refer the interested readers to references for further reading about these classifiers. For Lexicon-Sample WSD comparison I chose Htsa3[Grozea, 2004], a Trikhonov regularized least square classifier. For All-word WSD comparison, I chose IMS-Word2vec, a SVM-based classifier provided by Iacobacci et al [Iacobacci et al, 2016].

Performances All models mentioned above submitted their results in Senseval-3, which makes a fair comparison possible. I also chose the models training on same annotated corpus, Semcor, so that the results serve as a comparison of models' capabilities rather than data sets' qualities. For LSTM with label propagation in [Yuan et al, 2016], additional random 1000 unlabeled sentences per lemma were also used for training. The Model performances are summarized in Table 1 for Lexicon sample WSD and in Table 2 for All-words WSD.

Table 1: Lexicon Sample WSD performances in Senseval-3

Model	Training Corpus	Accuracy
MFS	N/A	55.2
Htsa3	Semcor	72.9
BLSTM	Semcor	73.4

Table 2: All-words WSD performances in Senseval-3

Model	Training Corpus	F1-Scores
MFS	N/A	62.3
ExtLesk	N/A	43.1
IMS+Word2Vec	Semcor	65.3
LSTM	Semcor	69.2
LSTM+LP	Semcor+1000	79.6

According to comparison Neural Networks model achieved state-of-art performance in both WSD tasks, showing the model superiority compared to other classifiers. For All-words WSD, using Label Propagation to enable semi-supervised learning also significantly improved model F1-scores, showing the validity of label propagation method.

6 Conclusions and future work

LSTM models, with its ability to capture long-term dependencies information, have shown promising results in both Lexicon sample WSD and All-words WSD. However several methods can still

possibly improve current LSTM models. First, the context embeddings used by Yuan et al were computed by plain averaging word embeddings. But as shown in [Ibacobacci et al, 2016], context embeddings computed by exponential decay which assigns more weight to contexts that are closer to target word could achieve better WSD performances. Second, the current LSTM model is a simple LSTM without attentions applied. With Label propagation, more complex LSTM should be trainable.

The metrics for evaluating WSD could also be reformed. In WSD, the most frequent sense baseline is strong because of the nature of human language. Models' abilities to correctly classify non-common sense should also be evaluated rather then simply report models' accuracy or F1-scores.

Current LSTM model doesn't leverage syntactic knowledge either. Neural Networks models such as Tree-RNN also have potential to be applied in knowledge-based WSD algorithms.

7 References

Chan, Y.S., Ng, H.T. and Chiang, D., 2007, June. Word sense disambiguation improves statistical machine translation. In *Annual Meeting-Association for Computational Linguistics* (Vol. 45, No. 1, p. 33).

Fellbaum, C. 1998. *WordNet: An Electronic Lexical Database.* Cambridge, MA: MIT Press.

Gale, W. A., Church, K., and Yarowsky, D. 1992b. A method for disambiguating word senses in a corpus. *Comput. Human.* 26, 415–439.

Grozea, C., 2004, July. Finding optimal parameter settings for high performance word sense disambiguation. In *Proceedings of Senseval-3 Workshop.*

Iacobacci, I., Pilehvar, M.T. and Navigli, R., 2016. Embeddings for word sense disambiguation: An evaluation study. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics* (Vol. 1, pp. 897-907).

Kågebäck, M. and Salomonsson, H., 2016. Word Sense Disambiguation using a Bidirectional LSTM. *arXiv* preprint arXiv:1606.03568.

Kilgarriff, A. 1998. SENSEVAL: An Exercise in Evaluating Word Sense Disambiguation Programs (1998). In *LREC*, CiteSeer.

Mikolov, T., Le, Q., and Sutskever, L. 2013. Exploiting similarities among languages for machine translation. *arXiv* preprint arXiv:1309.4168.

Miller, G., Leacock, C., Tengi, R., and Bunker, R. 1993. A semantic concordance. In *Proceedings of the ARPA Workshop on Human Language Technology.* 303–308.

Miller, G. 1995. WordNet: A Lexical Database for English. *Communications of the ACM* Vol. 38, No. 11: 39-41. Navigli, R., Litkowski, K., and Hargraves, C. 2007. Semeval-2007 task 07: Coarse-grained English all-words task. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, pages 30–35. Association for Computational Linguistics.

Navigili, R. 2009. Word sense disambiguation: A Survey. *ACM Comput. Surv.* 41,2, Article 10.

Ng, H., Yong, C., King, S. 1999. A Case Study on Inter-Annotator Agreement for Word Sense Disambiguation. *SINGLEX99.* pp 9-13.

Navigli, R. and Lapata, M., 2010. An experimental study of graph connectivity for unsupervised word sense disambiguation. *IEEE transactions on pattern analysis and machine intelligence*, 32(4), pp.678-692.

Pennington, J., Socher, R., and Manning, C. 2014. GloVe: Global Vectors for Word Representation. In *Proceedings of the 2014 EMNLP*, pages 1532–1543, Doha, Qatar.

Procter, P., Ed. 1978. *Longman Dictionary of Contemporary English.* Longman Group, Harlow, U.K.

Ravi, S. and Diao, Q., 2015. Large Scale Distributed Semi-Supervised Learning Using Streaming Approximation. *arXiv* preprint arXiv:1512.01752.

Stevenson, A., and Lindberg, C. 2010. *New Oxford American Dictionary.*

Stevenson, A. 2010. *Oxford Dictionary of English.*

Talukdar, P.P. and Crammer, K., 2009, September. New regularized algorithms for transductive learning. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases* (pp. 442-457). Springer Berlin Heidelberg.

Weaver, W. 1949. Translation. In *Machine Translation of Languages: Fourteen Essays* (written in 1949, published in 1955), W. N. Locke and A. D. Booth, Eds. Technology Press of MIT, Cambridge, MA, and John Wiley Sons, New York, NY, 15–23.

Yuan, D., Richardson, J., Doherty, R., Evans, C. and Altendorf, E., 2016. Semi-supervised Word Sense Disambiguation with Neural Models. *arXiv* preprint arXiv:1603.07012.