INDIAN INSTITUTE OF TECHNOLOGY, KANPUR

Word Sense Disambiguation using Localized RNNs

by

Rushab Munot

Roll Number: 14405

CS498A: Undergraduate Project II

under the supervision of

Dr. Harish Karnick

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

November 2017

Abstract

Word Sense Disambiguation using Localized RNNs

WSD addresses the task of differentiating the sense of a word depending upon its usage. We propose a localized end-to-end neural model at the word level with a word specific attention mechanism. We observe a decent performance with a word dependent but context independent attention profile for fixed size contexts. We also make use of the fact that WordNet tags are shared among words. This extra information is important when data is scarce (and it is generally scarce given the unavailability of large sense-tagged corpora). We evaluate this approach with specific words from the 'One million sense tagged instances' dataset and also the four-word dataset (interest, hard, line, serve) and English Lexical Sample Task from Senseval2. As future improvements, the model can be modified for generating sense vectors. One such approach would be modifying the WSD layer in the paper - 'Efficient Non-parametric Estimation of Multiple Embeddings per Word in Vector Space' (Neelakantan et Al., 2014).

Note: This work is a continuation of my UGP - 'Word Sense Disambiguation using RNNs for Context Embedding' under Prof. Harish Karnick, 2016-17/II. Many basic results and definitions have been taken from this Project.

Contents

Chapter 1

Introduction

A word can have multiple meanings when used differently. For example, while saying, "I love rock music" compared to "Igneous rocks are formed by the solidification of lava", the meaning of the word *rock* is not the same. In the first case in means a genre of music, while in the second case it refers to a geographic sense. They correspond the to following senses from WordNet3.1 [1].

1. S: (n) rock, stone (material consisting of the aggregate of minerals like those making up the Earth's crust) "that mountain is solid rock"; "stone is abundant in New England and there are many quarries"
2. S: (n) rock 'n' roll, rock'n'roll, rock-and-roll, rock and roll, rock, rock music (a genre of popular music originating in the 1950s; a blend of black rhythm-and-blues with white country-and-western) "rock is a generic term for the range of styles that evolved out of rock'n'roll."

As another example, consider the examples - "The surface is too hard to play on" and "The exam to hard to get through" which obviously have different meanings of the word *hard*.

The task of Word Sense Disambiguation (WSD) is to distinguish between senses of a single word depending upon its *context*.

The *context* around a word can be looked at as the immediate neighbourhood of the word.

Chapter 2

WordNet Senses

2.1 Developing a Hierarchy

Consider the following examples:

1. The new *set* of rules prohibits smoking on campus.
2. Can you please *set* the volume to a lower level?"

In the first sentence, the word *set* is used as a *noun* while in the second sentence it is used as a *verb*. Thus, it is sufficient to predict the Part-of-Speech (POS) of *set* while disambiguating these two senses. However, that may not always be sufficient.

Consider the following example, "The traitor set fire to the palace." Here too, *set* is used as a verb, but the meaning is "to start (a fire)" rather than "to adjust (the volume)". To disambiguate we need more information rather than just POS tags. This develops a hierarchical structure for prediction.

2.2 WordNet Senses

WordNet senses are stored in *Lexicographer Files* each of which have related words. For example, the file *noun.food* contains all nouns pertaining to food and drinks. There are 45 such files numbered from 00 to 44. WordNet represents senses with the help of sense keys. These are defined as follows (Theses definitions have been taken with reference from WordNet [1].Reference Link: https://wordnet.princeton.edu/man/senseidx.5WN.html)

$$\text{sense\_key} = \text{lemma} \ \% \ \text{lex\_sense}$$

lex_sense = ss_type:lex_filenum:lex_id:head_word:head_id

- **ss_type** is a one digit number between 1 and 5 which represents the *Synset* of the sense. Synset can be considered equivalent to basuc POS tags and are one of of *Noun[1], Verb[2], Adjective[3], Adverb[4], Adjective Satellite[5]*

- **lex_filenum** lies between 00 and 44 and is the is the lexicographer file number of the WordNet sense. Senses are categorized into Lexicographer files based on syntactic category and logical groupings. Some examples

  - File 04 represents nouns denoting acts or actions.
  - File 39 denotes perception related verbs. (seeing, hearing, feeling)

  Adjectives are represented within three files, nouns with in 26, verbs within 15 and adverbs in one file.

- **lex_id** is used to identify a sense within a lexicographer file. The lex_id and the lemma together provide a unique identity to every sense in the lexicographer file. Note that, there can be multiple senses of the same word in the same file.

- **head_word** and **head_id** are present in the sense_key only if the synset is Adjective Satellite i.e. ss_type = 5.
  Satellite adjectives are adjectives with a basic meaning which when appended with some context makes more sense. For example, *dry* which when appended with a context enhance the meaning.
  "dry" + "climate" = "arid"
  "thirsty" = "dry" + "throat"

- **head_id** with the head_word uniquely identifies the sense within a file, similar to lex_id.

Chapter 3

The Model

3.1 About Recurrent Neural Networks

Recurrent Neural Networks, typically Long Short-Term Memory Networks are used in Natural Language Processing to provide a basis for handling memory dependent relations. An LSTM network typically consists of the input gate which controls the effect of the input, the output gate decides what part of the stored state goes into the output and the forget gate which controls how much of the previous data is to be retained. (More details about LSTMs: http://colah.github.io/posts/2015-08-Understanding-LSTMs/)

3.2 A Localized attentive LSTM Model

The model is localized at the word level, that is one LSTM network per word. We train a Unidirectional (backward) LSTM model per word with a word specific attention profile. A fixed We observe that self attention mechanisms deviate around constant profile and the deviation is small. Thus, using a constant context-independent attention profile may help. The model is shown in figure **??**

If $h_1, \ldots, h_n$ denote the LSTM embedding from the LSTM networks over time, with a context window size of n, the context embedding is denoted by

$$context\_embeddingc = \sum_{n=1}^{N} a_i h_i$$

where a_i denote the attention weights.

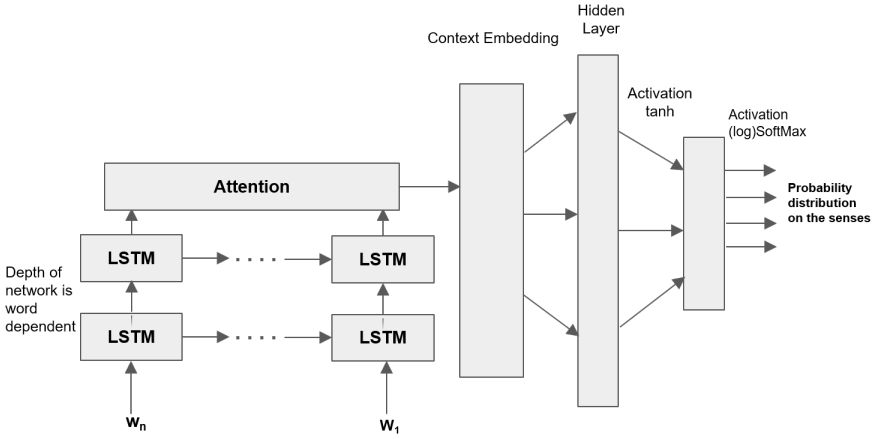

FIGURE 3.1: The Model and the Attention Mechanism
Image modified from my UGP titled "WSD using RNNs for context embedding" (CS396A, II Semester, 2016-17)[2]

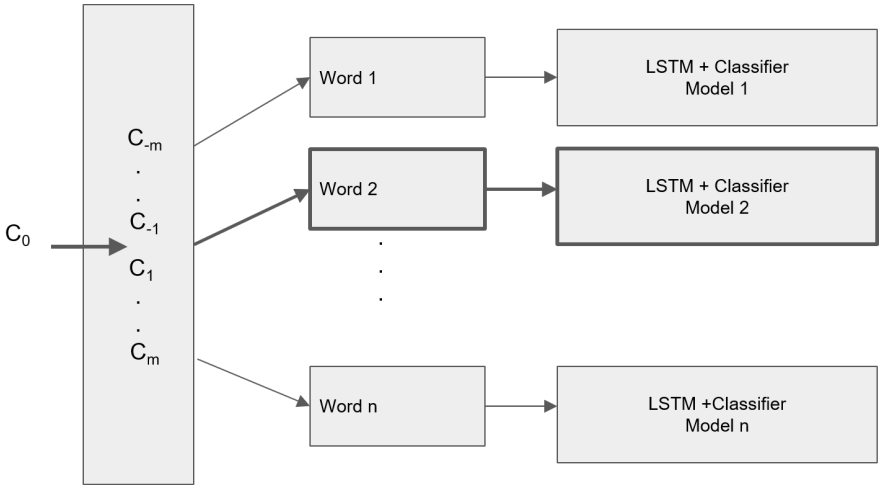

FIGURE 3.2: One model per word
Image taken from my UGP titled "WSD using RNNs for context embedding" (CS396A, II Semester, 2016-17)[2]

To emphasize the fact that these are word specific

$$c = \sum_{n=1}^{N} a_{w,i} h_i$$

Chapter 4

Datasets and Evaluation

4.1 Datasets

4.1.1 Senseval-2: Hard, Line, Serve, Interest datasets

In this dataset each of hard, line and serve have about 4000 examples while interest has about 200 examples. The distribution of senses can be found in the table given below.

Word	#Senses	Total # of examples	Distribution across senses
hard	3	4333	(3455, 502, 376)
serve	4	4378	(1814, 1272, 853, 439)
interest	6	2368	(1252, 500, 361, 178, 66, 11)
line	6	4146	(2217, 429, 404, 374, 373, 349)

TABLE 4.1: **Senseval-2 Four-Words Dataset [?]**
Taken from my UGP titled "WSD using RNNs for context embedding" (CS396A, II Semester, 2016-17)[2]

4.1.2 Senseval-2: Lexical Sample Task

In this dataset, we have very few sentences per word and a large number of senses. The statistics are given below

- Number of words = 73
- Number of sentences = 8611
- Average Number of words per sentence = 117
- Number of senses = 226 + 2 *(U, P)* = 228

- Number of senses per word =

Due to time constraints, we evaluate the results only on a few of these words. (See Results Section)

4.1.3 One Million Sense Tagged Instances Corpus

We evaluate our results on certain words from this corpus. The senses in this corpus are not tagged manually, but are semi-automatically tagged which may induce errors. The authors report an accuracy of 83.7% of the sense tags in the dataset

4.2 Evaluation Metric

Considering the highly imbalanced nature of the datasets, accuracy is not a good measure. F1 scores are better measures.

For the Four Word Dataset, the evaluation is done via a 10 fold-cross validation. However due to time constraints the models were stopped after 5 iteration in tne cross validation process.

For Senseval-2 English Lexical Sample Task we report the scores given by their system. (We get precision = recall for all words as we predict just one sense for every word, even when there can be multiple assignable senses)

4.3 Results

Word	**#Senses**	**BLSTM Embedding**		**Attention Mechanism**	
		Accuracy	**F1**	**Accuracy**	**F1**
hard	3	90.90	79.28	91.94	81.76
interest	4	87.67	80.42	90.95	86.12
serve	4	84.79	81.76	84.85	81.93
line	6	79.05	70.65	81.57	73.721

TABLE 4.2: Accuracy and F1 score on the 4 Words Dataset

Here we have predicted only one sense (instead of multiple senses). We have precision = recall and fine grained = coarse grained)

Word	**#Instances**	**#Senses**	**Score***
art	196	8	60.2
authority	184	9	62.5
bar	304	13	51.7
blind	108	9	80.0*
bum	92	6	75.6
chair	138	4	81.7*
channel	145	9	41.7
child	129	8	59.4
church	128	5	62.5
colorless	67	3	60.0*
cool	106	7	50.0
day	289	9	71.0

TABLE 4.3: Accuracy (as evaluated by their scorer)* on the Senseval-2, English Lexical Sample Task

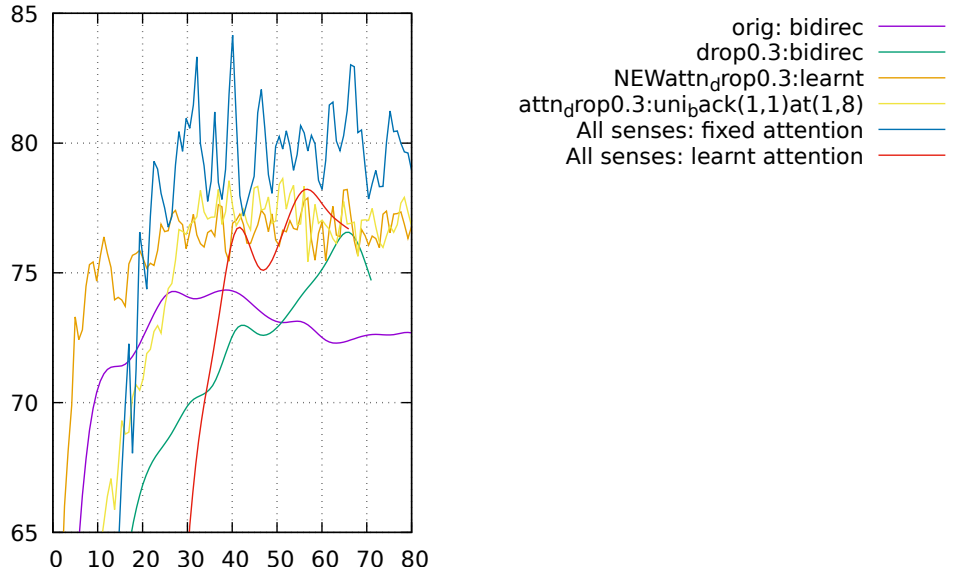

FIGURE 4.1: ***cause*: OMSTI dataset - Comparison between different methods**

4.3.1 Comparisons between different methods

4.4 Future Work

4.4.1 Exploiting the hierarchical nature of senses

In our results we have assumed that senses with the same combination in the $sense_key$ mean the same. While actually we have highly restricted our data. Instead one must consider all the senses in the same lexicographer file to be similar to a given sense. Thus, to predict the lexicographer file we can use all the senses in that file (this should be efective at least for nouns and verbs since they have a large number of lexicographer files). Once we have the lexicographer file number, another model trained specifically

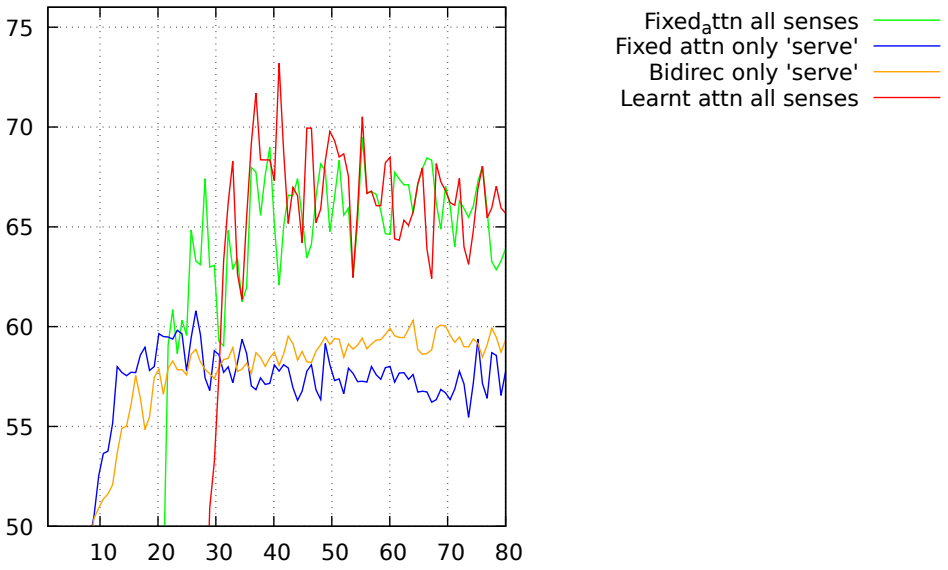

FIGURE 4.2: ***serve*: OMSTI dataset - Comparison between different methods**

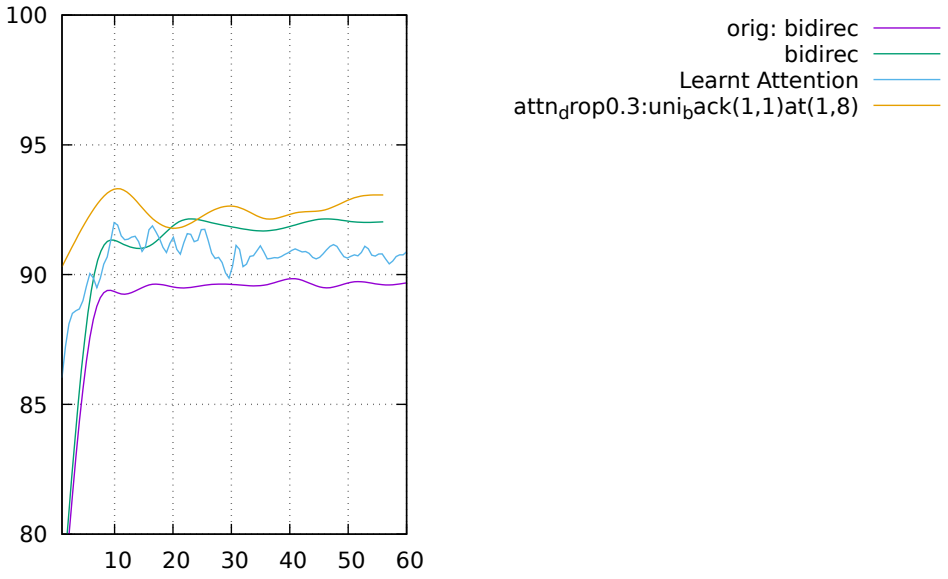

FIGURE 4.3: ***hard*: Four word dataset - Comparison between different methods**

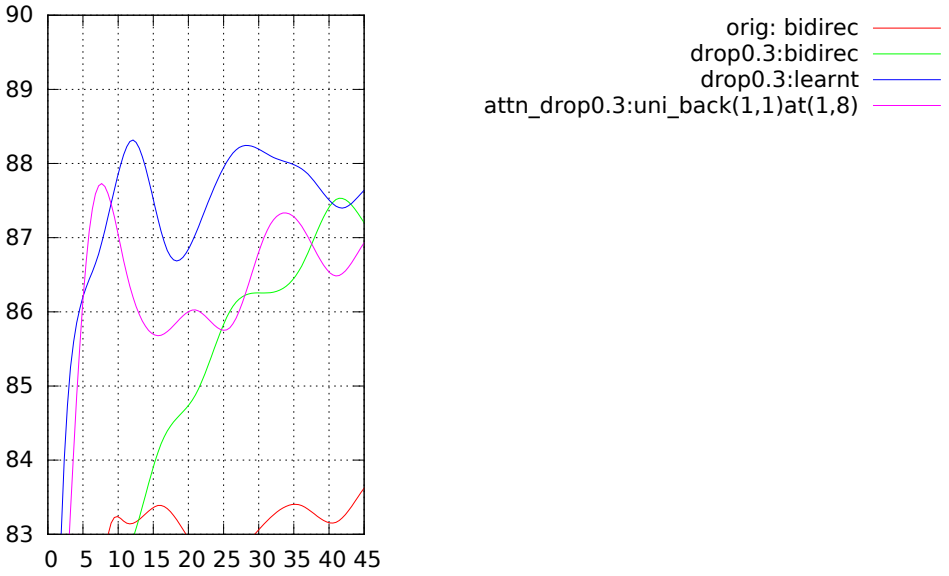

FIGURE 4.4: ***serve*: Four word dataset - Comparison between different methods**

for that lexicographer file can be used. Note that the lemma (which is the word to be disambiguated) is known. Also synsets are fairly easy to predict else another level of hierarchy could have been introduced.

4.4.2 Sense Vector Generation

Neelakantan et al [3] use context clustering to decide the relevant sense of the word being used, which can be thought of as a WSD layer. They embed this layer in to the skip-gram model. However, the context embeddings that they use are generated by averaging over the (global) word vectors of words in the context. Figure 4.5 depicts the same.

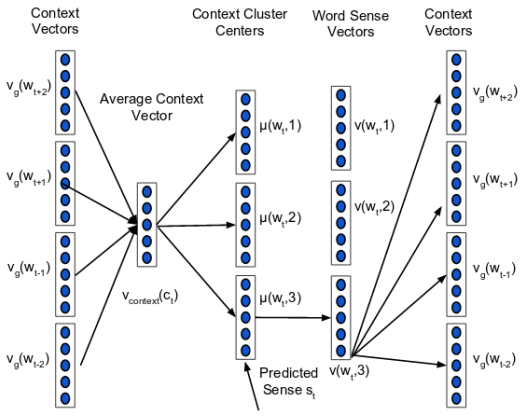

FIGURE 4.5: **Sense Vector Generation: Context Clustering approach of Neelakantan et al 2014**
Image taken from "Efficient non-parametricestimation of multiple embeddings per word in vector space. 2014", Neelakantan et al

Bibliography

[1] Princeton University. Wordnet. *"About WordNet." WordNet. Princeton University*, 2010. URL http://wordnet.princeton.edu.

[2] Harish Karkick Rushab Munot. Wsd using rnns for context embedding. *CS396A, Undergraduate Project I, IIT Kanpur*, 2016-17/II.

[3] Alexandre Passos Andrew McCallum Arvind Neelakantan, Jeevan Shankar. Efficient non-parametric estimation of multiple embeddings per word in vector space. 2014.

[4] Hans Salomonsson Mikael Kageback. Word sense disambiguation using a bidirectional lstm. *Proceedings of the Workshop on Cognitive Aspects of the Lexicon*, 2016.

[5] Kaveh Taghipour and Hwee Tou Ng. One million sense tagged instances for word sense disambiguation and induction. *Proceedings of the 19th Conference on Computational Language Learning, pages 338344, Beijing, China, July 30-31, 2015*, 2015.

[6] Quoc V. Le Ilya Sutskever, Oriol Vinyals. Sequence to sequence learning with neural networks. 2014.

[7] Cristopher Olah. Understanding lstm networks. *Colah's blog*, 2015.

[8] R. Navigli. Word sense disambiguation: A survey. 2009.

[9] Sepp Hochreiter Juergen Schmidhuber. Long short-term memory. *Neural computation, Vol. 9, pages 1735-1780*, 1997.