# Word Sense Disambiguation using RNNs for Context Embedding

CS396A - UGP Presentation

Rushab Munot (14405)
CSE, IITK

**Supervisor**
Prof. Harish Karnick
CSE, IITK

April 28, 2017

# Word Sense Disambiguation

- A given word may have multiple meanings depending on the context that they appear in.

## Example

1. The exam was hard and students barely managed to pass.
2. I have no hard feelings for you.
3. The track was too hard for a morning jog.

# Word Sense Disambiguation

- A given word may have multiple meanings depending on the context that they appear in.

**Example**

1. The exam was hard and students barely managed to pass.
2. I have no hard feelings for you.
3. The track was too hard for a morning jog.

- Depending on the context each instance of the word *hard* has a different meaning. The task of WSD is to identify which sense a given context refers to.

**Example**

A more subtle example - *public interest vs self-interest*

1. The decision was taken in the interest of the majority.
2. It was not in her interest to perform the allocated task.

# What is the context?

- When we say that the meaning depends on the context what do we mean?

# What is the context?

- When we say that the meaning depends on the context what do we mean?
- Typical approaches use a local window around the word to be disambiguated.

**Example**

The customers were assured by {*the bank that* **interest** *rates for deposits*} would soon increase.

# What is the context?

- When we say that the meaning depends on the context what do we mean?
- Typical approaches use a local window around the word to be disambiguated.

**Example**

The customers were assured by $\{$*the bank that* **interest** *rates for deposits*$\}$ would soon increase.

- Use a bag of words from this window $\Rightarrow$ $\{$'the', 'bank', $\ldots$, 'deposits'$\}$.

# What is the context?

- When we say that the meaning depends on the context what do we mean?
- Typical approaches use a local window around the word to be disambiguated.

**Example**

The customers were assured by {*the bank that* **interest** *rates for deposits*} would soon increase.

- Use a bag of words from this window ⇒ {'the', 'bank', . . . , 'deposits'}.
- Add sequential information by assigning weights to the words ⇒{ ('the',-3), ('bank', -2), ('the', -1), ('rates',1), ('for', 2), ('deposits', 3) } Note that the weights can be a function of the position, {('the', w(-3), . . . ,('deposits', w(3)} where w is a weight function.

# Context-Embeddings - Add more information*

- Information like Part-of-Speech (POS) tags can be added to words in context

Taken from *Word Sense Disambiguation: A Survey* (R. Navigli, 2009)

# Context-Embeddings - Add more information*

- Information like Part-of-Speech (POS) tags can be added to words in context
- Use chunking i.e. divide the sentence into groups of words which form a meaningful syntactic formation, for eg. phrases (noun, verb, ...)

Taken from *Word Sense Disambiguation: A Survey* (R. Navigli, 2009)

# Context-Embeddings - Add more information*

- Information like Part-of-Speech (POS) tags can be added to words in context
- Use chunking i.e. divide the sentence into groups of words which form a meaningful syntactic formation, for eg. phrases (noun, verb, ...)
- Parsing the sentence to extract a syntactic representation of the sentence.

Taken from *Word Sense Disambiguation: A Survey* (R. Navigli, 2009)

# Context-Embeddings - Add more information*

- Information like Part-of-Speech (POS) tags can be added to words in context
- Use chunking i.e. divide the sentence into groups of words which form a meaningful syntactic formation, for eg. phrases (noun, verb, ...)
- Parsing the sentence to extract a syntactic representation of the sentence.
- Add knowledge based information to context using an external thesaurus like Word-Net.

Taken from *Word Sense Disambiguation: A Survey* (R. Navigli, 2009)

# Classical Approaches*

Supervised Approaches

- Represent the context and use a classifier (Naive Bayes, SVM, kNN, Decision Tree, Feed Forward NNs and Ensemble Methods*)

Taken from *Word Sense Disambiguation: A Survey* (R. Navigli, 2009)

# Classical Approaches*

Supervised Approaches

- Represent the context and use a classifier (Naive Bayes, SVM, kNN, Decision Tree, Feed Forward NNs and Ensemble Methods*)
- SVMs have better accuracy than others.

Taken from *Word Sense Disambiguation: A Survey* (R. Navigli, 2009)

# Classical Approaches*

Supervised Approaches

- Represent the context and use a classifier (Naive Bayes, SVM, kNN, Decision Tree, Feed Forward NNs and Ensemble Methods*)
- SVMs have better accuracy than others.
- Semi-supervised approaches using BootStrapping

Taken from *Word Sense Disambiguation: A Survey* (R. Navigli, 2009)

# Classical Approaches*

Supervised Approaches

- Represent the context and use a classifier (Naive Bayes, SVM, kNN, Decision Tree, Feed Forward NNs and Ensemble Methods*)
- SVMs have better accuracy than others.
- Semi-supervised approaches using BootStrapping
- Unsupervised Approaches

Taken from *Word Sense Disambiguation: A Survey* (R. Navigli, 2009)

# Classical Approaches*

Supervised Approaches

- Represent the context and use a classifier (Naive Bayes, SVM, kNN, Decision Tree, Feed Forward NNs and Ensemble Methods*)
- SVMs have better accuracy than others.
- Semi-supervised approaches using BootStrapping

- Unsupervised Approaches
- Context-clustering: Represent the cluster as a context vector from the co-occurence matrix, reduce dimensionality, and clusterize these vectors. While testing assign the cluster with highest similarity.

Taken from *Word Sense Disambiguation: A Survey* (R. Navigli, 2009)

# Classical Approaches*

Supervised Approaches

- Represent the context and use a classifier (Naive Bayes, SVM, kNN, Decision Tree, Feed Forward NNs and Ensemble Methods*)
- SVMs have better accuracy than others.
- Semi-supervised approaches using BootStrapping

- Unsupervised Approaches
- Context-clustering: Represent the cluster as a context vector from the co-occurence matrix, reduce dimensionality, and clusterize these vectors. While testing assign the cluster with highest similarity.
- Using weighted co-occurence graphs

Taken from *Word Sense Disambiguation: A Survey* (R. Navigli, 2009)

# Classical Approaches*

- Knowledge-based methods: Use a thesaurus/dictionary (e.g. WordNet) to add more information.

Taken from *Word Sense Disambiguation: A Survey* (R. Navigli, 2009)

# Classical Approaches*

- Knowledge-based methods: Use a thesaurus/dictionary (e.g. WordNet) to add more information.
- Lesk's Algorithm Similarity defined as the overlap between the *gloss* of the context and word. $gloss(w)$ represents (bag of) words in the definitions of $w$. $gloss(context(w))$ represents union of *glosses* of all words in context.

$$Similarity_{Lesk} = |gloss(w) \cap gloss(context(w))|$$

Taken from *Word Sense Disambiguation: A Survey* (R. Navigli, 2009)

# Classical Approaches*

- Knowledge-based methods: Use a thesaurus/dictionary (e.g. WordNet) to add more information.
- Lesk's Algorithm Similarity defined as the overlap between the *gloss* of the context and word. $gloss(w)$ represents (bag of) words in the definitions of $w$. $gloss(context(w))$ represents union of *glosses* of all words in context.

$$Similarity_{Lesk} = |gloss(w) \cap gloss(context(w))|$$

- Many more approaches with several measures of similarity.

Taken from *Word Sense Disambiguation: A Survey* (R. Navigli, 2009)

# Using RNNs

- Used extensively for tasks on sequential data.

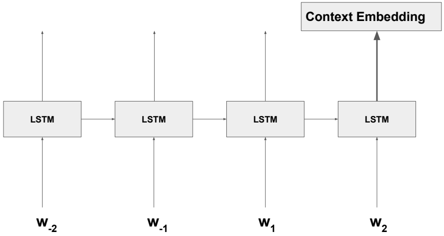


Figure: Context Embedding using LSTMs

# Using RNNs

- Used extensively for tasks on sequential data.

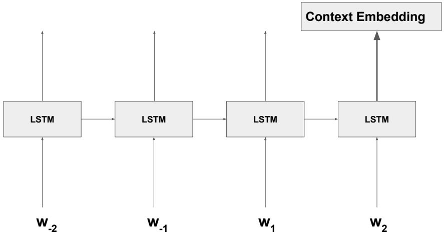


Figure: Context Embedding using LSTMs

- The sequential information is inherent in such models.

- Used extensively for tasks on sequential data.

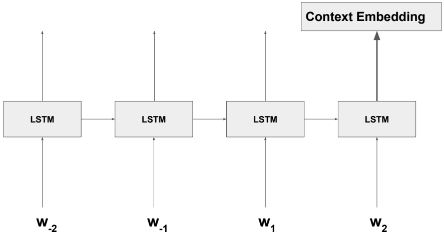


Figure: Context Embedding using LSTMs

- The sequential information is inherent in such models.
- Words can be passed as word vectors. (word2vec - Mikolov et al. 2013, glove - Socher et al. 2014)

# Using RNNs

- Used extensively for tasks on sequential data.

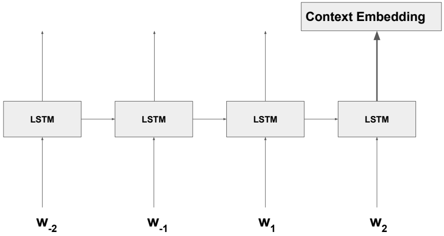


Figure: Context Embedding using LSTMs

- The sequential information is inherent in such models.
- Words can be passed as word vectors. (word2vec - Mikolov et al. 2013, glove - Socher et al. 2014)
- After training the model (somehow) for the context embedding, train a classifier which maps the context embedding to the sense labels.

# The Model

- View the problem as a classification problem on the sense labels.

# The Model

- View the problem as a classification problem on the sense labels.
- Train an end-to-end model, mapping the one hot vectors of the words in the context to the sense label. ¡3-¿

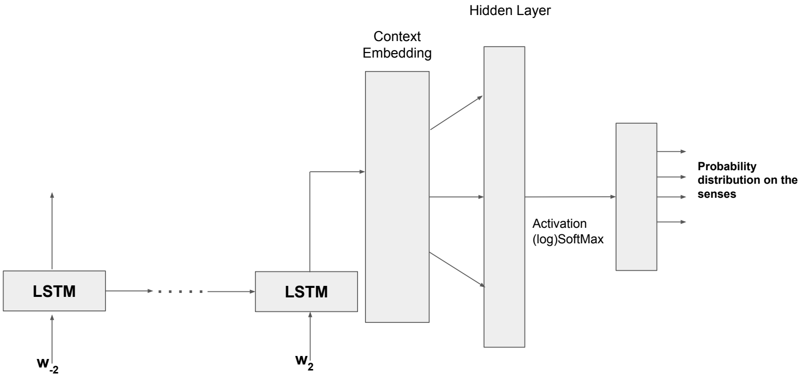


Figure: Training a sense classifier using LSTMs

# The Model

- Scan the sentence in both directions i.e. use a bidirectional LSTM.

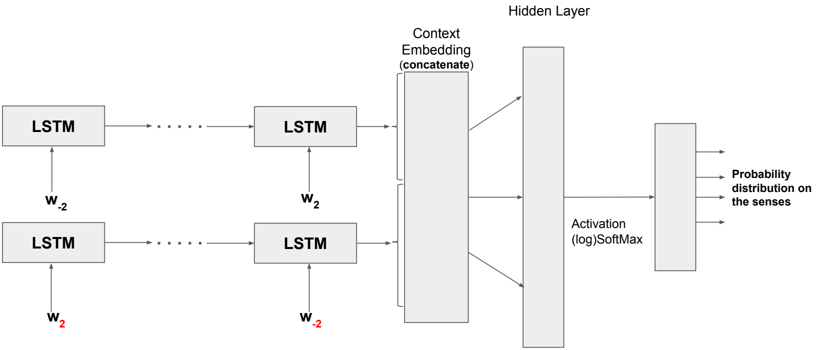


Figure: Training a sense classifier: Bidirectional LSTM

# The Model

- Scan the sentence in both directions i.e. use a bidirectional LSTM.
- Some dependencies seem to be resolved by this.

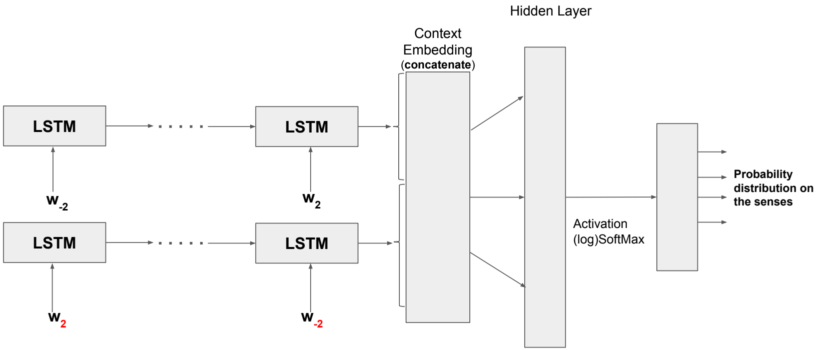


Figure: Training a sense classifier: Bidirectional LSTM

# The Model - One LSTM per Word

- Train a classifier for every word that is to be disambiguated.

# The Model - One LSTM per Word

- Train a classifier for every word that is to be disambiguated.
- Thus if we have three words $w_1$, $w_2$ and $w_3$ with, say, 3, 4, 5 senses respectively, we train three classifiers, with 3, 4, 5 classes respectively.

# The Model - One LSTM per Word

- Train a classifier for every word that is to be disambiguated.
- Thus if we have three words $w_1$, $w_2$ and $w_3$ with, say, 3, 4, 5 senses respectively, we train three classifiers, with 3, 4, 5 classes respectively.
- Thus, we build a local model for every word instead of a shared model for all words.

# Adding More Information

- More information like POS tags, AMR representation, etc. can be added to improve the model

# Adding More Information

- More information like POS tags, AMR representation, etc. can be added to improve the model
- The POS tags can be appended to the words and passed as $<$word, POS$>$ tuples.

# Adding More Information

- More information like POS tags, AMR representation, etc. can be added to improve the model
- The POS tags can be appended to the words and passed as <word, POS> tuples.
- The POS tags can be separately passed through another unidirectional/bidirectional LSTM and then appended to the context vector.

# Datasets

- Senseval-2 dataset of 4 words - hard, line, interest, serve

| 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: Senseval-2 Four-Words Dataset

- Certain words from the One-million word corpus interest = 4 senses ( 500 each), position = 5 senses ( 500 each), serve = 8 senses (6 senses  500, remaining 2 have lesser instances)

# Accuracy

| Word | #Senses | BLSTM Embedding | | BLSTM + POS embedding | |
|---|---|---|---|---|---|
| | | Accuracy | F1 | Accuracy | F1 |
| hard | 3 | 90.90 | 79.28 | 91.04 | 79.36 |
| interest | 4 | 87.67 | 80.42 | 87.24 | 80.65 |
| serve | 4 | 84.79 | 81.76 | 84.14 | 81.09 |
| line | 6 | 79.05 | 70.65 | 78.64 | 68.53 |

Table: Accuracy and F1(macro) score on the Senseval-2 dataset

- Accuracy not a good measure, as data is unbiased (for Senseval-2), F1 is a better measure

# Accuracy

| Word | #Senses | BLSTM Embedding | | BLSTM + POS embedding | |
|---|---|---|---|---|---|
| | | **Accuracy** | **F1** | **Accuracy** | **F1** |
| interest | 4 | 76.61 | 76.96 | 76.40 | 72.20 |
| position | 5 | 61.39 | 60.50 | 62.86 | 61.76 |
| serve | 8 | 53.29 | 54.32 | 54.64 | 50.00 |

Table: Accuracy and F1(macro) score on the million-word dataset

# Effect of POS Tags

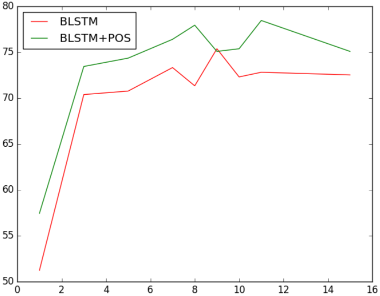


Figure: For the word *interest** (one-million word corpus)

* Specifically chosen example to show the effect, such large variance may not be seen, at times no increase in accuracy is also observed. Note that this must be tested on exactly same data as changing the data may greatly affect results

# Problems

- Using a SVM based classifier or a one-vs-rest Logistic Regression.

# Problems

- Using a SVM based classifier or a one-vs-rest Logistic Regression.
- Gives slightly better results (1-2%), but the C parameter (in sklearn) needs to be fine-tuned.

# Problems

- Using a SVM based classifier or a one-vs-rest Logistic Regression.
- Gives slightly better results (1-2%), but the C parameter (in sklearn) needs to be fine-tuned.
- However this value turns out to be data dependent, and changes with different words as also (in some cases) with different runs for the same word.

# Problems

- Insufficient data - difference between accuracy with 60% training data and 90% training data is significant

# Problems

- Insufficient data - difference between accuracy with 60% training data and 90% training data is significant
- Skewed data - The actual instances that go into the training set and test set largely affect the accuracies (as much as 8-9% in certain cases)

# Problems

- Insufficient data - difference between accuracy with 60% training data and 90% training data is significant
- Skewed data - The actual instances that go into the training set and test set largely affect the accuracies (as much as 8-9% in certain cases)
- There is no guarantee that if a model works on a certain dataset, it will work in general.

# Problems

- Insufficient data - difference between accuracy with 60% training data and 90% training data is significant
- Skewed data - The actual instances that go into the training set and test set largely affect the accuracies (as much as 8-9% in certain cases)
- There is no guarantee that if a model works on a certain dataset, it will work in general.
- Accuracy high on some words low on others. This has been observed in the literature too.

# Problems

- Insufficient data - difference between accuracy with 60% training data and 90% training data is significant
- Skewed data - The actual instances that go into the training set and test set largely affect the accuracies (as much as 8-9% in certain cases)
- There is no guarantee that if a model works on a certain dataset, it will work in general.
- Accuracy high on some words low on others. This has been observed in the literature too.
- For words equal number of examples (one million words corpus) in each class, F1 $\approx$ Accuracy, however more data is needed to correctly classify the senses as 500 examples of each sense are not enough.

# Future Work

- Build a structural hierarchy of senses, say a tree where the topmost node represents the given word and as we go down every node represents a cluster of senses. The classification goes in a top-down manner like a B+ tree.

# Future Work

- Build a structural hierarchy of senses, say a tree where the topmost node represents the given word and as we go down every node represents a cluster of senses. The classification goes in a top-down manner like a B+ tree.
- Try constructing a dataset, which can guarantee universally valid results. That is a model when tested on this dataset can be said to work on most other datasets.

# Future Work

- Build a structural hierarchy of senses, say a tree where the topmost node represents the given word and as we go down every node represents a cluster of senses. The classification goes in a top-down manner like a B+ tree.
- Try constructing a dataset, which can guarantee universally valid results. That is a model when tested on this dataset can be said to work on most other datasets.
- Instead of a classifier use regression models more extensively.

# Acknowledgements


- Datasets from Senseval-2 and One million word corpus (Kaveh Taghipour and Hwee Tou Ng, 2015 - One Million Sense-Tagged Instances for Word SenseDisambiguation and Induction)
- Word Sense Disambiguation: A Survey - R. Navigli, 200
- Word Sense Disambiguation using a Bidirectional LSTM (Mikael Kageback, Hans Salomonsson, 2016)

THANK YOU