Word Sense Disambiguation

EE391A - UGP Presentation

Shanu Kumar (150659)[1]

Supervisors

Prof. Harish Karnick [2] Prof. K S Venkatesh [1]

[1]Electrical Engineering, IITK

[2]Computer Science and Engineering, IITK

April 11, 2018

Word Sense Disambiguation

- Word sense disambiguation (WSD) is the ability to identify the meaning of words in context.

Example

- Banks have increased interest rates.
- It was in my interest to do so.
- Match was interesting.

Word Sense Disambiguation

- Word sense disambiguation (WSD) is the ability to identify the meaning of words in context.

Example

- Banks have increased interest rates.
- It was in my interest to do so.
- Match was interesting.

Example

- In the peoples interest - public-interest
- My interest in the field - self-interest.

Word Sense Disambiguation

- Word sense disambiguation (WSD) is the ability to identify the meaning of words in context.

Example

- Banks have increased interest rates.
- It was in my interest to do so.
- Match was interesting.

Example

- In the peoples interest - public-interest
- My interest in the field - self-interest.

- So, the sense of the word can be interpreted in multiple ways depending upon the context where they appears.

Word Specific Model

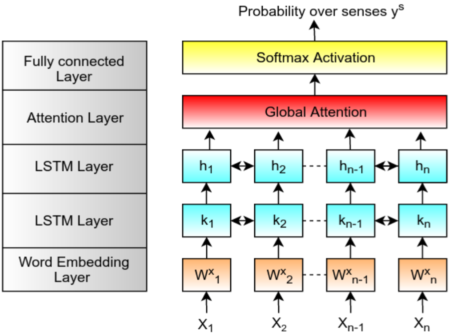

Figure: Word specific model

- Model is different for every word to be disambiguated.

Word Specific Model

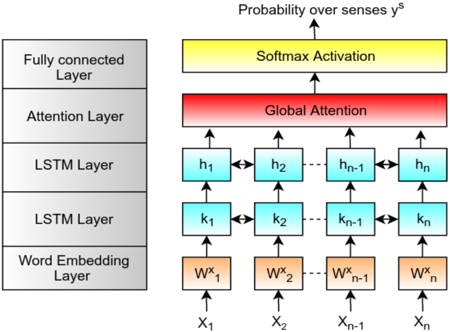

Figure: Word specific model

- Model is different for every word to be disambiguated.
- Attention mechanism computes the context vector depending upon the target word.

Word Specific Model

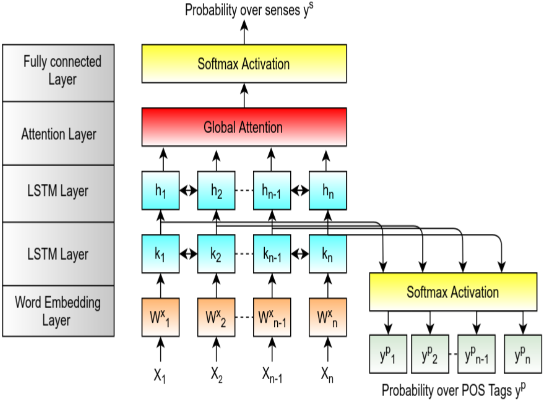

Figure: Word specific model+POS Tags

- Tried to improve the hidden states of the word by predicting POS tags of every word.

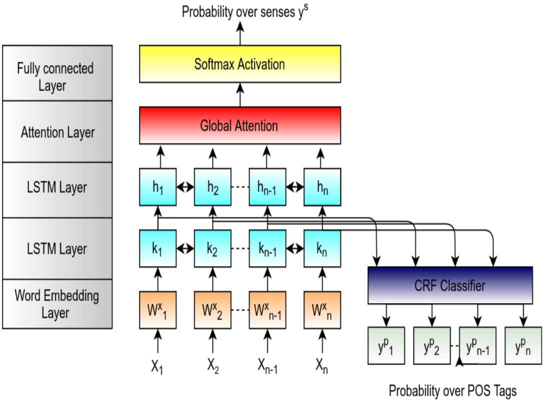

Figure: Word specific model+POS Tags+CRF

- Used Conditional Random Fields (CRF) for improving the accuracy of POS tags.

WordNet Senses

- To disambiguate a word, we only need certain no. of characteristics.
- For example, to disambiguate the word set in the following examples

Example

- Set the volume. - verb
- Set of rules. - noun

To distinguish btw these two senses, we only need to know the POS tags.

WordNet Senses

- To disambiguate a word, we only need certain no. of characteristics.
- For example, to disambiguate the word set in the following examples

Example

- Set the volume. - verb
- Set of rules. - noun

To distinguish btw these two senses, we only need to know the POS tags.

- However, to disambiguate between two verb senses of the same word, we need more informative tags

Example

- To set fire - Verb $\Longrightarrow$ To start a fire
- To set the volume - Verb $\Longrightarrow$ To adjust the volume

These two senses cannot be distinguished only with POS tags

Word Specific Hierarchical Model

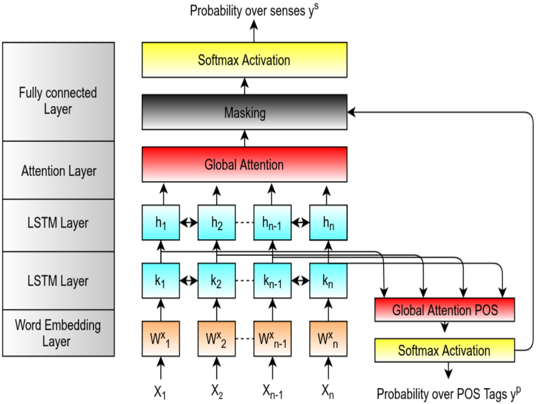

Figure: Word specific hierarchical model

- Using WordNet Senses, we develop a hierarchical model, in where we first predict POS tags which suppresses the other senses for classiying

Word Specific Datasets

We use the Senseval-2 Lexical Sample Task dataset for the Word specific model.

Word	#Senses	# 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

Word Specific Datasets

We only trained on these words: open, force, make, point, support, serve, place on One-million word corpus.

Word	#Senses	# of examples	Distribution across Senses
serve	4	3421	(1941#V1, 839#V2, 529#V3, 112#V4)
place	6	3511	(1149#N1, 623#V1, 490#V2, 488#N2, 479#V3, 282#N3)
make	7	6566	(2006#V1, 1025#V2, 968#V3, 962#V4, 617#V5, 543#N6, 445#V7)
open	5	2913	(990#ADJ1, 662#V1, 632#V2, 565#V3, 64#ADJ2)
support	7	3423	(1020#V1, 670#N1, 533#V2, 503#V3, 470#V4, 170#V5, 57#N2)
force	5	3649	(1150#N1, 969#N2, 543#V1, 495#N3, 492#N4)
point	8	2766	(989#N1, 518#V1, 479#N2, 282#N3, 193#N4, 163#N5, 87#V2, 55#V3)

Table: One Million Dataset: SemCor+OMSTI

Word Specific Results

Target Word	F1-Score			Accuracy		
	Train	Val	Test	Train	Val	Test
Hard	89.45%	78.66%	78.11%	94.85%	89.37%	89.78
Serve	95.75%	89.80%	89.84%	96.49%	95.5%	91.94
Interest	84.32%	80.50%	72.33%	92.06%	89.06%	86.16
Line	87.98%	82.45%	78.73%	92.08%	88.75%	86.33

Table: Senseval-2 Four-Words Dataset Results

Word Specific Results

Sense Word	Model	F1 Score Train	F1 Score Val	Accuracy Train	Accuracy Val
Force	Model-1	98.42%	91.49%	98.40%	92.01%
	Model-2	97.21%	89.74%	97.36%	90.97%
	Model-3	97.24%	90.26%	97.39%	91.49%
	Model-4	97.65%	89.33%	97.74%	90.62%
Make	Model-1	75.21%	49.33%	75.87%	51.91%
	Model-2	65.62%	50.44%	66.72%	52.34%
	Model-3	67.33%	52.59%	68.65%	54.08%
Open	Model-1	95.72%	77.30%	96.31%	82.98%
	Model-2	93.53%	77.88%	94.05%	84.03%
	Model-3	92.87%	78.35%	94.31%	84.37%
	Model-4	94.38%	76.60%	94.62%	84.55%

Table: Results on One Million Dataset: SemCor+OMSTI

Word Specific Results

Sense Word	Model	F1 Score Train	F1 Score Val	Accuracy Train	Accuracy Val
Place	Model-1	96.32%	83.58%	96.65%	84.89%
	Model-2	93.29%	81.68%	94.01%	83.33%
	Model-3	94.30%	83.99%	94.98%	85.07%
	Model-4	93.69%	83.24%	94.31%	84.03%
Point	Model-1	94.58%	75.63%	96.35%	83.85%
	Model-2	91.27%	73.87%	93.89%	83.59%
	Model-3	92.52%	75.60%	94.60%	83.85%
	Model-4	92.52%	75.81%	94.74%	84.63%
Serve	Model-1	90.40%	79.57%	90.96%	82.12%
Support	Model-1	90.63%	68.51%	90.25%	67.01%
	Model-2	86.47%	72.75%	85.30%	71.00%
	Model-3	87.65%	69.00%	87.05%	68.92%
	Model-4	88.80%	63.30%	88.76%	64.93%

Table: Results on One Million Dataset: SemCor+OMSTI

All-words Model

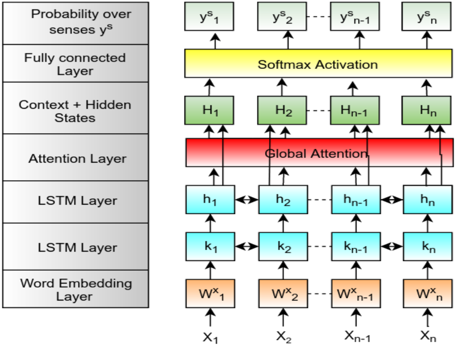

Figure: All-words model

- Instead of framing a separate classification problem for each given word, this models the joint disambiguation of the target text as a whole in terms of a sequence labeling.

All-words Model

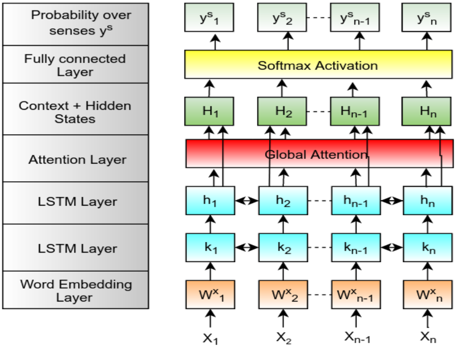

Figure: All-words model

- Instead of framing a separate classification problem for each given word, this models the joint disambiguation of the target text as a whole in terms of a sequence labeling.
- Here we compute a context vector for the senetnce using Attention.

All-words Model

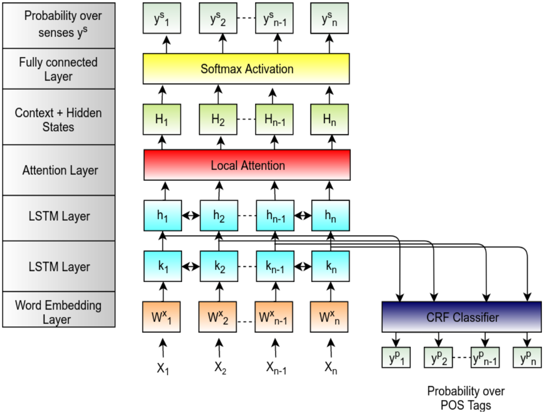

Figure: All-words model+Local Attention

- Most of the time we only need a window around the target word for predicting its sense. Instead of computing context vector for the

All-words Model

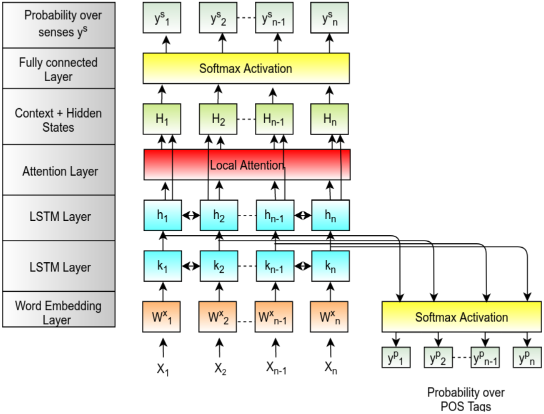

Figure: All-words model+Local Attention+Hidden States

- Concatenated both context vector and the hidden states for every word

All-words Model

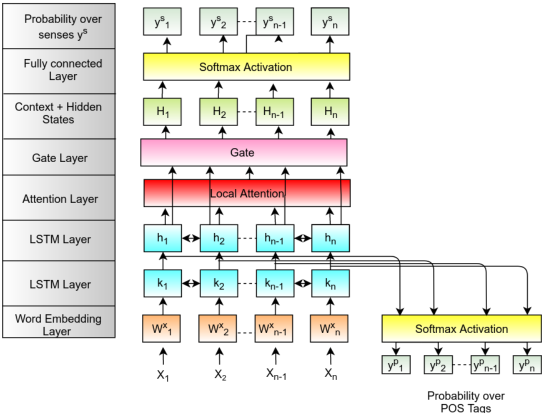

Figure: All-words model+Gated Attention

- If model can learn when to use context vector and hidden states or their combination.

All-words Model

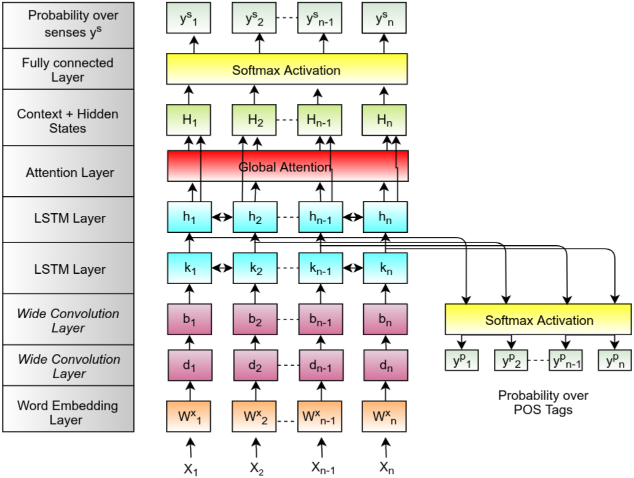

Figure: All-words model+CNN

All-words Model+CNN

- Convolutional neural networks (CNN) extract local features using local window around a word.
- It works like local attention but applied on word vectors instead of hidden states.
- This model outperform all the previous models

All-words Hierarchical Model

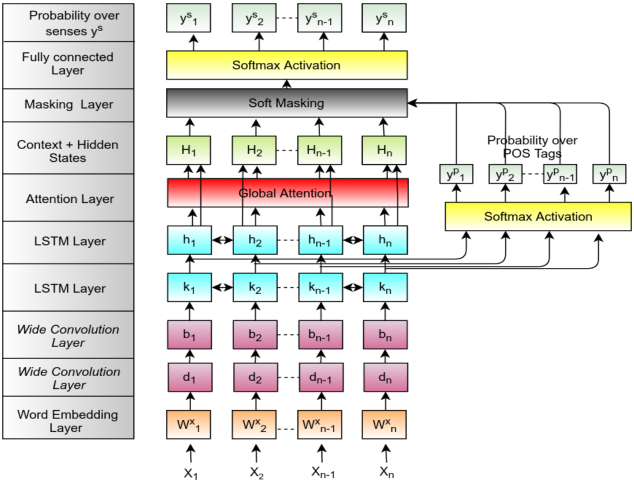

Figure: All-words Hierarchical Model

All-words Hierarchical Model

We also tried the hierarchical model after success in the Word specific model. Here we used two variants of masking technique after predicting the POS Tags for every word in the sentence.

All-words Hierarchical Model

We also tried the hierarchical model after success in the Word specific model. Here we used two variants of masking technique after predicting the POS Tags for every word in the sentence.

Hierarchical Model+Soft Masking

Here we simply multiplied the probabilities of the corresponding POS tags with the probabilities of the senses using WordNet. This model outperform the All-word Model with CNN.

All-words Hierarchical Model

We also tried the hierarchical model after success in the Word specific model. Here we used two variants of masking technique after predicting the POS Tags for every word in the sentence.

Hierarchical Model+Soft Masking

Here we simply multiplied the probabilities of the corresponding POS tags with the probabilities of the senses using WordNet. This model outperform the All-word Model with CNN.

Hierarchical Model+Hard Masking

Used similar masking technique to the Word specific model. Suppose a word is predicted to be noun, then all senses which are not noun are added a negative value to suppress their probabilities, thus this masking technique is hard.

All-words Model Results

Table 2 shows the performance of models trained on the One million sentences corpus consisting of 680066 sentences and 45 classes of senses. These scores are evaluated on the same dataset consisting of 170016 sentences.

Model	F1-Score	Accuracy
All-word Model	65.54%	73.16%
All-word Model+PT+Local Attention	44.36%	53.75%
All-word Model+PT+Local Attention+Hidden States	52.19%	58.68%
All-word Model+PT+Gated Attention*	44.17%	53.07%
All-word Model+PT+Local Attention+Hidden States+CRF	50.65%	57.15%
All-word Model+PT+CNN	72.33%	77.93%
All-word Hierarchical Model+Soft Masking	74.04%	79.38%
All-word Hierarchical Model+Hard Masking*	70.35%	77.30%

Table: Results on One Million Dataset: SemCor+OMSTI, * shows that these models are early stopped

Conclusion

- We defined, analyzed and compared experimentally different end-to-end models of varying complexities, including different variants of attention, masking mechanisms.
- Unlike the word specific model, where a dedicated model needs to be trained for every target word and each disambiguation target is treated in isolation, all-words model learn a single model in one pass from the training data, and then disambiguate jointly all target words within an input text.
- Hierarchical models outperform in both the models. Hence we can say that hierarchical models are the key for WSD task.
- The use of POS Tags only improved the context vector but its effect on the accuracy is negligible.
- Convolutional neural networks (CNN) extracts local features around a word, what actually humans do for disambiguating senses.

Future Work

- We plan to extend the hierarchical model from only POS Tags to POS Tag $\rightarrow$ lexical num $\rightarrow$ lexical id using WordNet database.
- The model can be used for sense vector generation.

Thank You