A Systematic Study of Neural Discourse Models for Implicit Discourse Relation

Attapol T. Rutherford. Vera Demberg Nianwen Xue

Presenter: Dhruv Agarwal

INTRODUCTION

- Inferring implicit discourse relations is a difficult subtask in discourse parsing.
- Typical approaches have used hand crafted features from the two arguments and suffer from data sparsity problems.
- Neural network approaches need to be applied on small datasets and possess no common experimental settings for evaluation.
- This paper conducts several experiments to compare various neural architectures in literature and publishes their results.

DISCOURSE

- High level organization of text can be characterized as discourse relations between adjacent pairs of texts.
- There are two types of discourse relations:
    - **Explicit Discourse Relations**
    - **Implicit Discourse Relations**

EXPLICIT DISCOURSE

According to Lawrence Eckenfelder, a securities industry analyst at Prudential-Bache Securities Inc., "Kemper is the first firm to make a major statement with program trading." He added that ***"having just one firm do this isn't going to mean a hill of beans****. But* ***if this prompts others to consider the same thing, then it may become much more important****."*

The discourse connective is '**but**', and the sense is **Comparison.Concession**

IMPLICIT DISCOURSE

*According to Lawrence Eckenfelder, a securities industry analyst at Prudential-Bache Securities Inc., **"Kemper is the first firm to make a major statement with program trading**." He added **that "having just one firm do this isn't going to mean a hill of beans**. But if this prompts others to consider the same thing, then it may become much more important."*

The omitted discourse connective is '**however**'.

and the sense is **Comparison.Contrast**.

CHALLENGES

- Predicting internal discourse relations is fundamentally a semantic task and relevant semantics might be difficult to recover from surface level features.

Bob gave Tina the burger.

She was hungry.

- Purely vector based representations of arguments might not be sufficient to capture discourse relations.

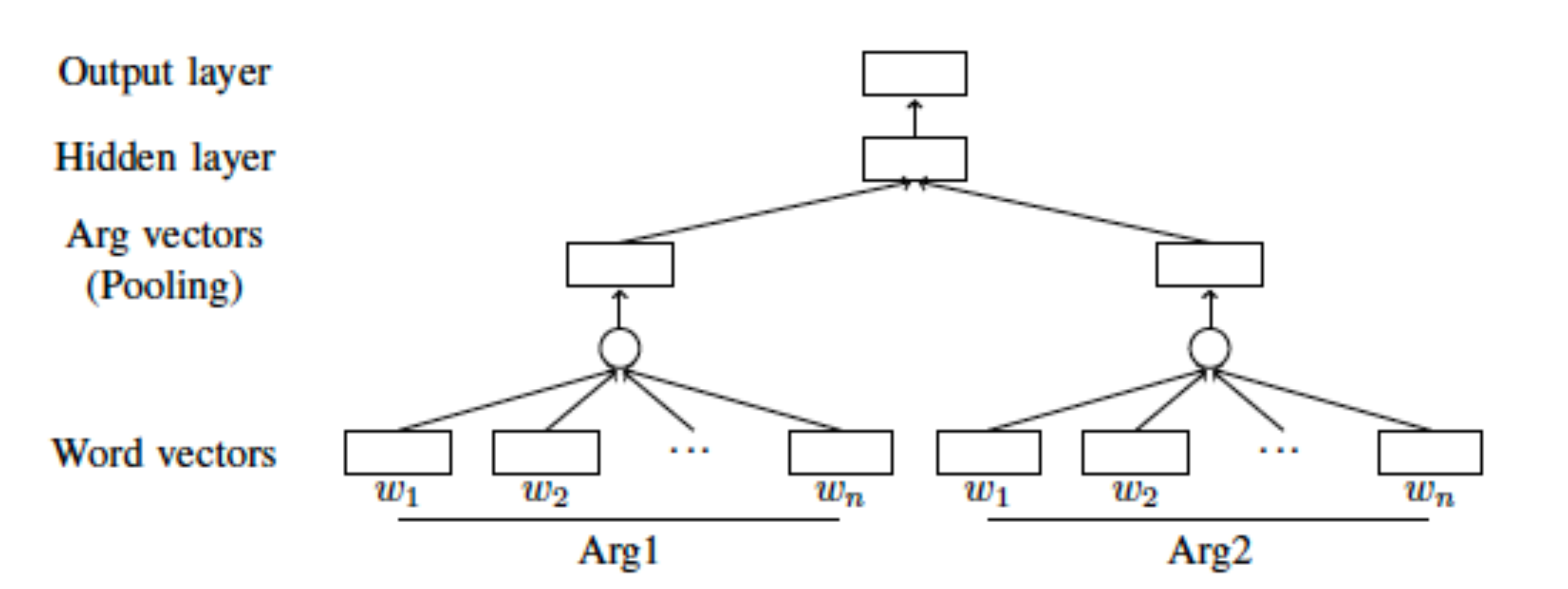
Bob gave Tina the burger.

He was hungry.

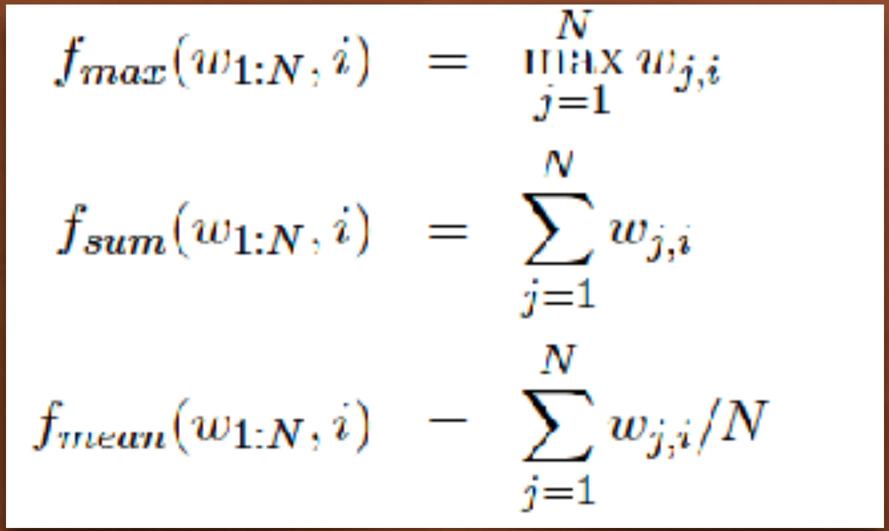
MODEL ARCHITECTURES

- In order to find the best distributed representation and network architecture , they explore by probing the different points on the spectrum of structurality from structureless bag-of-words models to sequential and tree-structured models.
    - Bag of words Feed Forward Model
    - Sequential LSTM
    - Tree LSTM

FEED FORWARD MODEL



THREE KINDS OF POOLING ARE CONSIDERED: MAX, MEAN AND SUMMATION AS FOLLOWS,

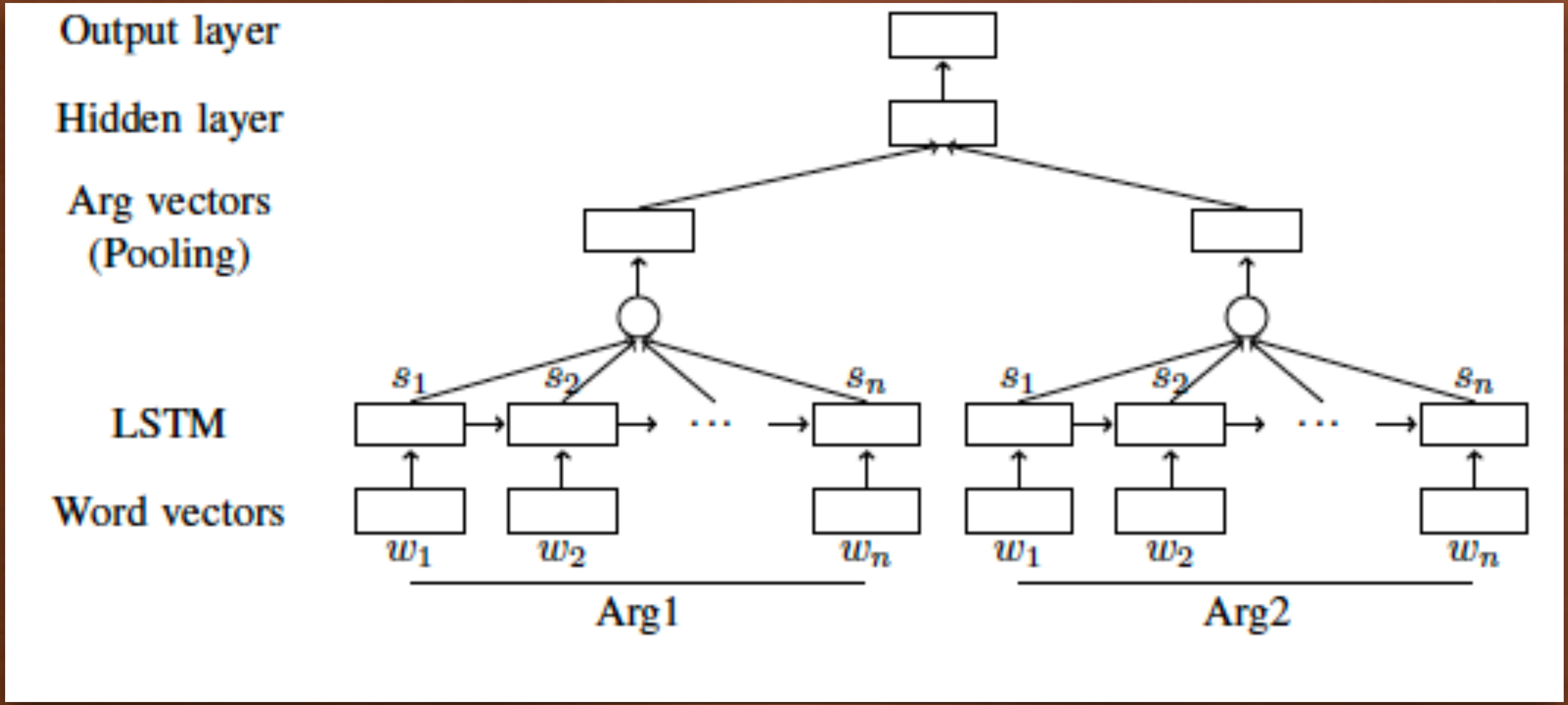


$$f_{max}(w_{1:N}, i) = \max_{j=1}^{N} w_{j,i}$$

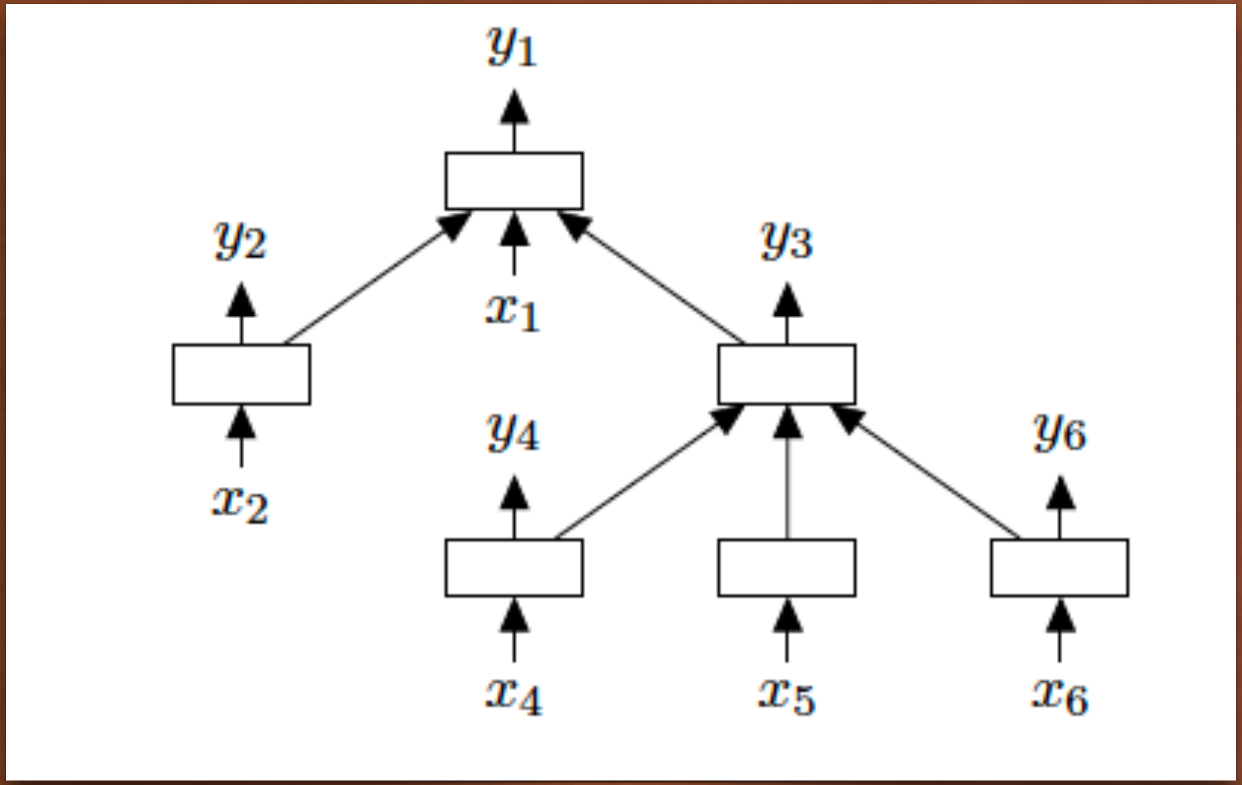
$$f_{sum}(w_{1:N}, i) = \sum_{j=1}^{N} w_{j,i}$$

$$f_{mean}(w_{1:N}, i) = \sum_{j=1}^{N} w_{j,i}/N$$

SEQUENTIAL LSTM



TREE LSTM

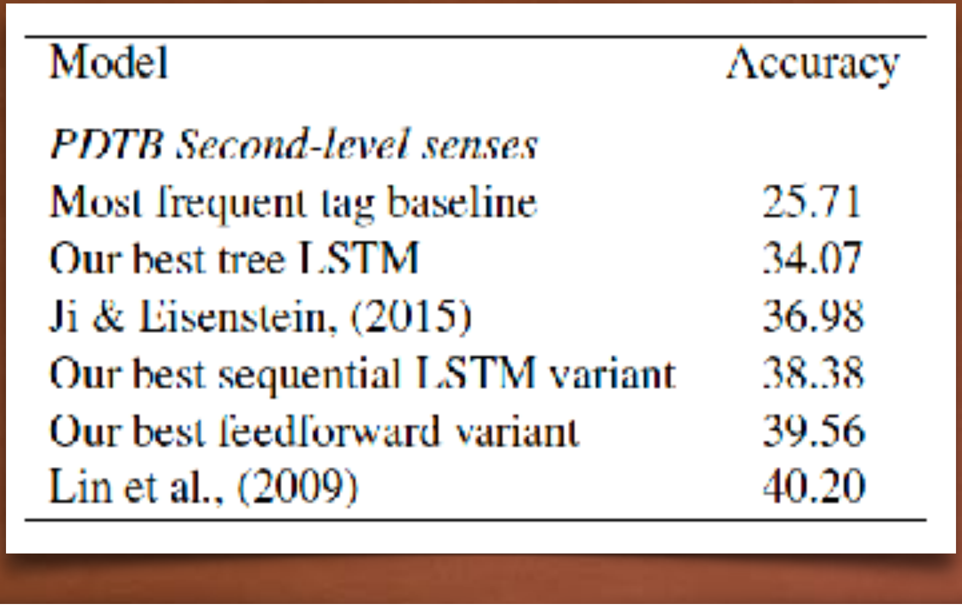


- THE DIFFERENCE BETWEEN STANDARD LSTM AND TREE LSTM IS THAT GATING VECTORS AND MEMORY CELL UPDATES ARE BASED ON HIDDEN STATES OF MANY CHILD NODES.
- THE HIDDEN STATE VECTOR CORRESPOND TO A CONSTITUENT IN THE TREE.

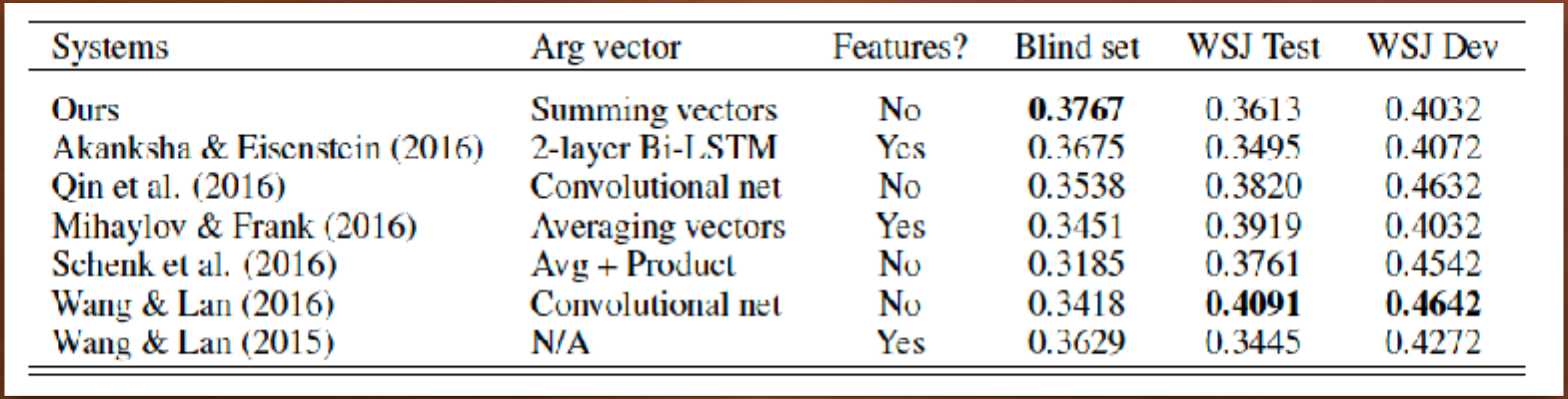
IMPLEMENTATION DETAILS

- Penn Discourse Tree Bank is used because of its theoretical simplicity and large size.
- The PDTB provides three levels of discourse relations, each level providing finer semantic distinctions.
- The task is carried out on the second level with 11 classes.
- Cross Entropy Loss function, Adagrad Optimizer and no regularization/dropout.
- The model performance is also evaluated on CONLL shared task 2015 and CDTB.

RESULTS

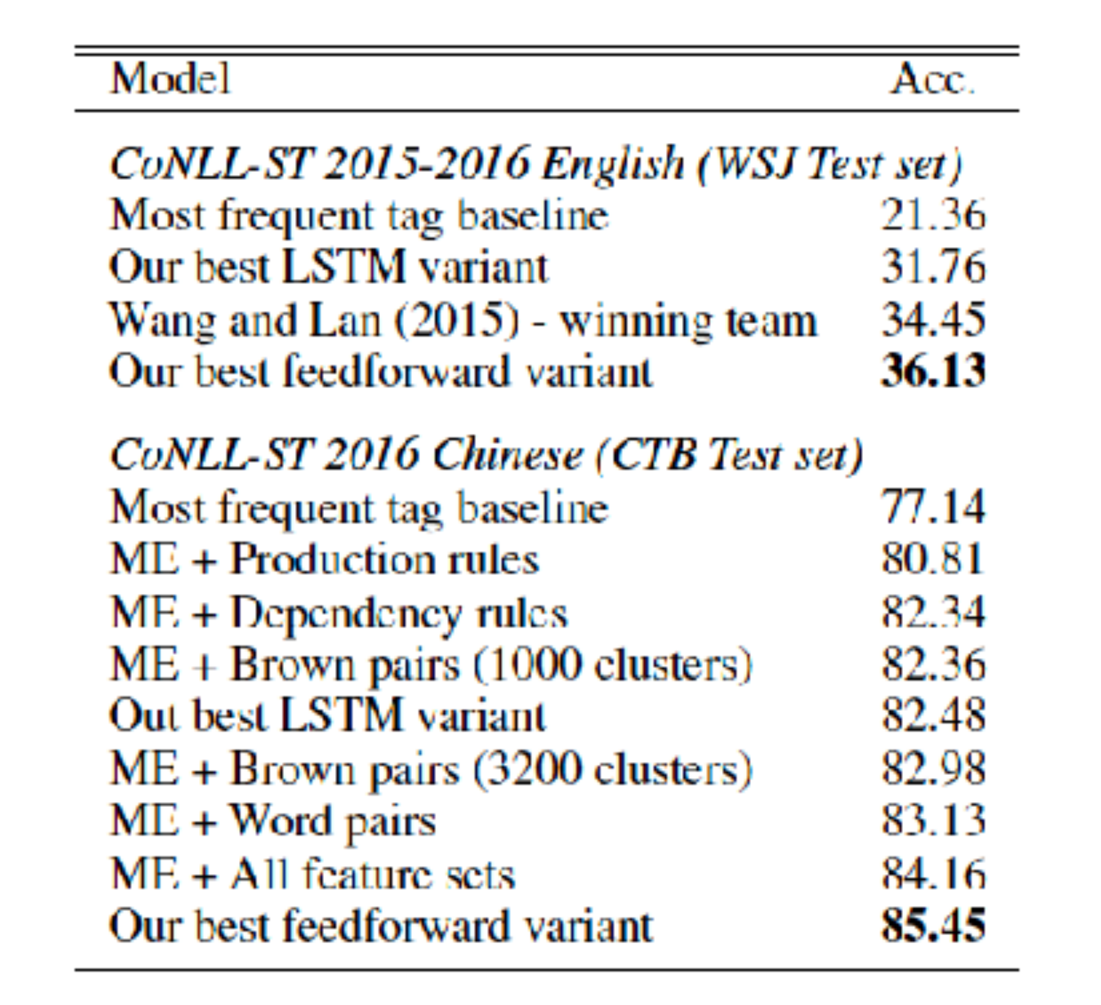


Model	Accuracy
PDTB Second-level senses	
Most frequent tag baseline	25.71
Our best tree LSTM	34.07
Ji & Eisenstein, (2015)	36.98
Our best sequential LSTM variant	38.38
Our best feedforward variant	39.56
Lin et al., (2009)	40.20



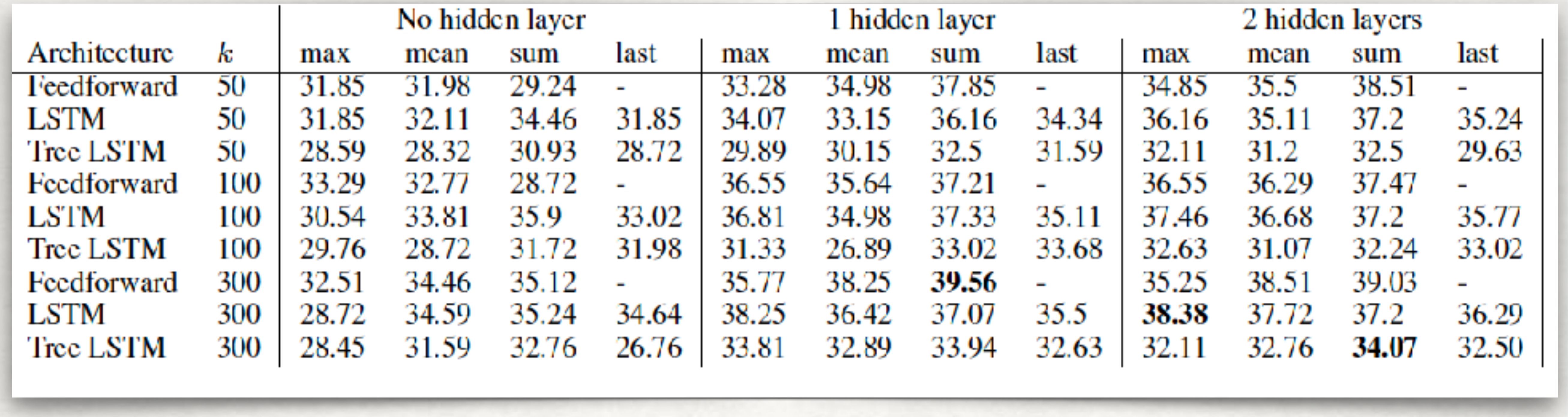
Systems	Arg vector	Features?	Blind set	WSJ Test	WSJ Dev
Ours	Summing vectors	No	**0.3767**	0.3613	0.4032
Akanksha & Eisenstein (2016)	2-layer Bi-LSTM	Yes	0.3675	0.3495	0.4072
Qin et al. (2016)	Convolutional net	No	0.3538	0.3820	0.4632
Mihaylov & Frank (2016)	Averaging vectors	Yes	0.3451	0.3919	0.4032
Schenk et al. (2016)	Avg + Product	No	0.3185	0.3761	0.4542
Wang & Lan (2016)	Convolutional net	No	0.3418	**0.4091**	**0.4642**
Wang & Lan (2015)	N/A	Yes	0.3629	0.3445	0.4272

- FEEDFORWARD MODEL IS THE BEST OVERALL AMONG ALL THE NEURAL ARCHITECTURES THEY EXPLORE.
- IT OUTPERFORMS LSTM BASED,CNN BASED AND THE BEST MANUAL SURFACE FEATURE BASED MODELS IN SEVERAL SETTINGS.



Model	Acc.
CoNLL-ST 2015-2016 English (WSJ Test set)	
Most frequent tag baseline	21.36
Our best LSTM variant	31.76
Wang and Lan (2015) - winning team	34.45
Our best feedforward variant	**36.13**
CoNLL-ST 2016 Chinese (CTB Test set)	
Most frequent tag baseline	77.14
ME + Production rules	80.81
ME + Dependency rules	82.34
ME + Brown pairs (1000 clusters)	82.36
Out best LSTM variant	82.48
ME + Brown pairs (3200 clusters)	82.98
ME + Word pairs	83.13
ME + All feature sets	84.16
Our best feedforward variant	**85.45**

- For Baseline comparison Max Entropy models are used which are loaded with feature sets such as dependency rule pairs, production rule pairs and Brown Cluster pairs.



		No hidden layer				1 hidden layer				2 hidden layers			
Architecture	k	max	mean	sum	last	max	mean	sum	last	max	mean	sum	last
Feedforward	50	31.85	31.98	29.24	-	33.28	34.98	37.85	-	34.85	35.5	38.51	-
LSTM	50	31.85	32.11	34.46	31.85	34.07	33.15	36.16	34.34	36.16	35.11	37.2	35.24
Tree LSTM	50	28.59	28.32	30.93	28.72	29.89	30.15	32.5	31.59	32.11	31.2	32.5	29.63
Feedforward	100	33.29	32.77	28.72	-	36.55	35.64	37.21	-	36.55	36.29	37.47	-
LSTM	100	30.54	33.81	35.9	33.02	36.81	34.98	37.33	35.11	37.46	36.68	37.2	35.77
Tree LSTM	100	29.76	28.72	31.72	31.98	31.33	26.89	33.02	33.68	32.63	31.07	32.24	33.02
Feedforward	300	32.51	34.46	35.12	-	35.77	38.25	**39.56**	-	35.25	38.51	39.03	-
LSTM	300	28.72	34.59	35.24	34.64	38.25	36.42	37.07	35.5	**38.38**	37.72	37.2	36.29
Tree LSTM	300	28.45	31.59	32.76	26.76	33.81	32.89	33.94	32.63	32.11	32.76	**34.07**	32.50

- Sequential LSTMs outperform feedforward model when word vectors are not high dimensional and not trained on a larger corpus.

- Summation pooling is effective for both LSTM and feedforward models, since word vectors are known to have additive properties.

DISCUSSION

- Sequential and Tree LSTM might work better if they had a larger amount of annotated data.
- Benefits of Tree LSTM cannot be realized if the model discards syntactic categories in intermediate nodes.
- Linear interaction allows combination of high dimensional vectors without exponential growth of parameters.

CONCLUSION

- Manually crafted surface features are not important for this task and it holds true for different languages.

- Expressive power of distributed representations can overcome data sparsity issues of traditional approaches.

- Simple feed-forward architecture can outperform more sophisticated architectures such as sequential and tree-based LSTM networks, given the **small amount of data.**

- The paper compiles the results of all the previous systems and provides a common experimental setting for future research.

THANK YOU