Hierarchical Attention Networks for Document Classification

Zichao Yang, Diyi Yang, Chris Dyer, Xiadong He, Alex Smola, Eduard Hovy

Presented by Volodymyr Medentsiy

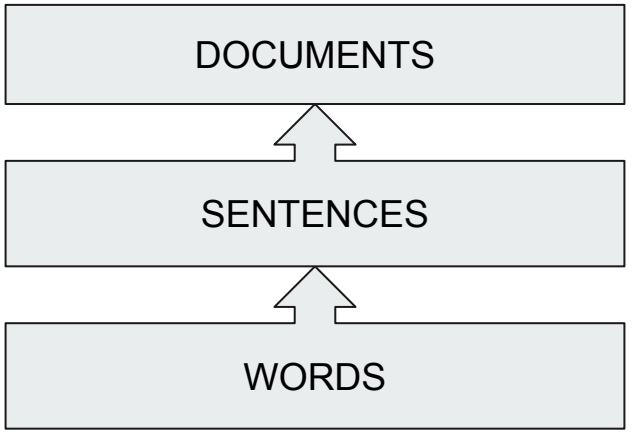
Plan

1. Key contributions
2. Model architecture
3. Experiments and analysis
4. Conclusion

Motivation

Observation 1:
Documents have hierarchical structure

Result:
Construct document embedding in a hierarchical manner



Motivation

Observation 2:
Not all words and sentences are born equal

Result:
Use attention at word and sentence level

pork belly = delicious . || scallops? || I don't even like scallops, and these were a-m-a-z-i-n-g . || fun and tasty cocktails. || next time I in Phoenix, I will go back here. || Highly recommend.

Figure: Highlighted parts deliver stronger meaning

Hierarchical Attention Network

Combined these insights to build HAN model.

Key idea is to apply the following algorithm on the sentence and word level:

1. run encoder network on the sequence of words (or sentences)
2. use attention to highlight relevant components

Model architecture

Building blocks of the model. Word level

1. Map words w_{it} to word embeddings x_{it} (word2vec, Glove)
2. Run BiGRU to get contextual word annotations $[\overrightarrow{h_{it}}, \overleftarrow{h_{it}}]$
3. Apply Attention mechanism to get s_i - the sentence representation

$$
\begin{aligned}
\mathrm{x}_{it} &= W_e w_{it} \\
\overrightarrow{h_{it}} &= \overrightarrow{GRU}(x_{it}) \\
\overleftarrow{h_{it}} &= \overleftarrow{GRU}(x_{it}) \\
\mathrm{h}_{it} &= [\overrightarrow{h_{it}}, \overleftarrow{h_{it}}] \\
\mathrm{u}_{it} &= tanh(W_w h_{it} + b_w) \\
\alpha_{it} &= \frac{exp(u_{it}^T u_w)}{\sum_t exp(u_{it}^T u_w)} \\
\mathrm{s}_i &= \sum_t \alpha_{it} h_{it}
\end{aligned}
$$

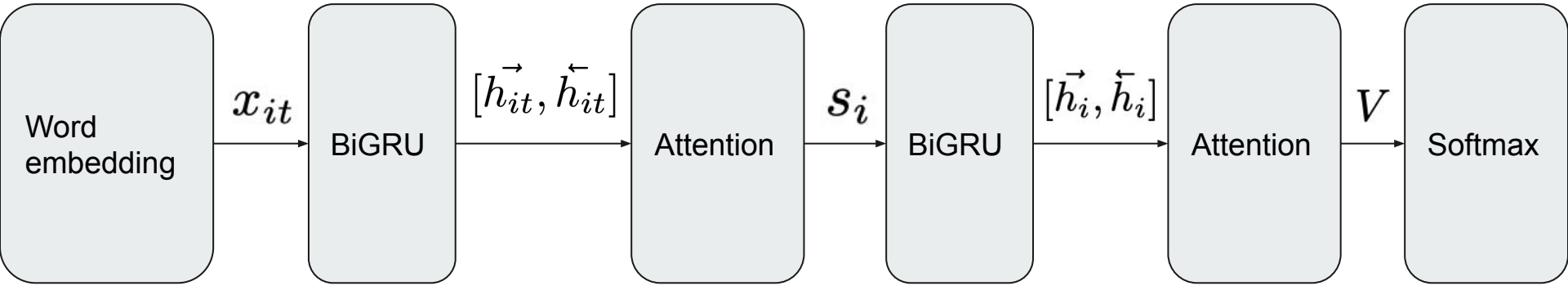
Model architecture

Building blocks of the model. Sentence level

1. Given sentence embeddings s_i run the same algorithm for sentences
2. Run BiGRU to get contextual sentence annotations $[\vec{h_i}, \overleftarrow{h_i}]$
3. Apply Attention mechanism to get the document embedding V

$$\vec{h_i} = \vec{GRU}(s_i)$$
$$\overleftarrow{h_i} = \overleftarrow{GRU}(s_i)$$
$$\mathrm{h}_i = [\vec{h_i}, \overleftarrow{h_i}]$$
$$\mathrm{u}_i = tanh(W_s h_i + b_w)$$
$$\alpha_i = \frac{exp(u_i^T u_s)}{\sum_i exp(u_i^T u_s)}$$
$$\mathrm{v} = \sum_t \alpha_i h_i$$

Model architecture



Data

Use 6 data sets for experiments:

1.Sentiment estimation

- Yelp 2013, 2014, 2015 reviews. Ratings from 1 to 5
- IMDB reviews. Ratings from 1 to 10
- Amazon reviews. Ratings from 1 to 5

2. Topic classification

- Yahoo answers. 10 classes of topics

Experiments

Data

80% training , 10% validation, 10% test

Data set	classes	documents	average #s	max #s	average #w	max #w	vocabulary
Yelp 2013	5	335,018	8.9	151	151.6	1184	211,245
Yelp 2014	5	1,125,457	9.2	151	156.9	1199	476,191
Yelp 2015	5	1,569,264	9.0	151	151.9	1199	612,636
IMDB review	10	348,415	14.0	148	325.6	2802	115,831
Yahoo Answer	10	1,450,000	6.4	515	108.4	4002	1,554,607
Amazon review	5	3,650,000	4.9	99	91.9	596	1,919,336

Baseline models

1. Linear models with document statistics as features
    a. BOW, BOW+TFiDF
    b. n-grams, n-grams+TFiDF
    c. Bag-of-means
2. SVM
    a. Text features
    b. AverageSG
    c. SSWE

Baseline models

3. Neural models
    a. CNN-word, CNN-char
    b. LSTM
    c. Conv-GRNN, LSTM-GRNN

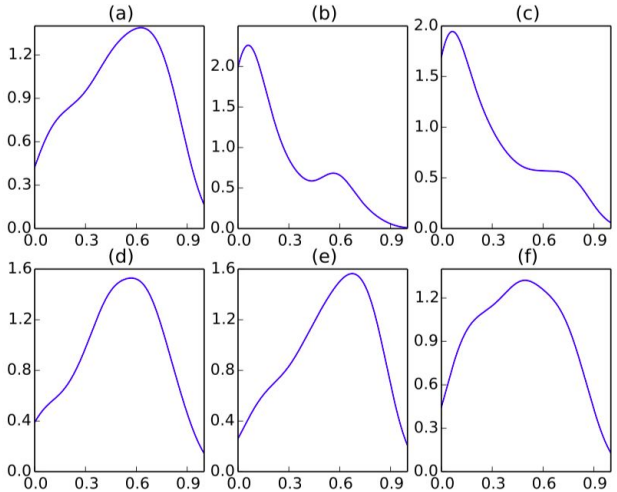
Experiments

	Methods	Yelp'13	Yelp'14	Yelp'15	IMDB	Yahoo Answer	Amazon
Zhang et al., 2015	BoW	-	-	58.0	-	68.9	54.4
	BoW TFIDF	-	-	59.9	-	71.0	55.3
	ngrams	-	-	56.3	-	68.5	54.3
	ngrams TFIDF	-	-	54.8	-	68.5	52.4
	Bag-of-means	-	-	52.5	-	60.5	44.1
Tang et al., 2015	Majority	35.6	36.1	36.9	17.9	-	-
	SVM + Unigrams	58.9	60.0	61.1	39.9	-	-
	SVM + Bigrams	57.6	61.6	62.4	40.9	-	-
	SVM + TextFeatures	59.8	61.8	62.4	40.5	-	-
	SVM + AverageSG	54.3	55.7	56.8	31.9	-	-
	SVM + SSWE	53.5	54.3	55.4	26.2	-	-
Zhang et al., 2015	LSTM	-	-	58.2	-	70.8	59.4
	CNN-char	-	-	62.0	-	71.2	59.6
	CNN-word	-	-	60.5	-	71.2	57.6
Tang et al., 2015	Paragraph Vector	57.7	59.2	60.5	34.1	-	-
	CNN-word	59.7	61.0	61.5	37.6	-	-
	Conv-GRNN	63.7	65.5	66.0	42.5	-	-
	LSTM-GRNN	65.1	67.1	67.6	45.3	-	-
This paper	HN-AVE	67.0	69.3	69.9	47.8	75.2	62.9
	HN-MAX	66.9	69.3	70.1	48.2	75.2	62.9
	HN-ATT	**68.2**	**70.5**	**71.0**	**49.4**	**75.8**	**63.6**

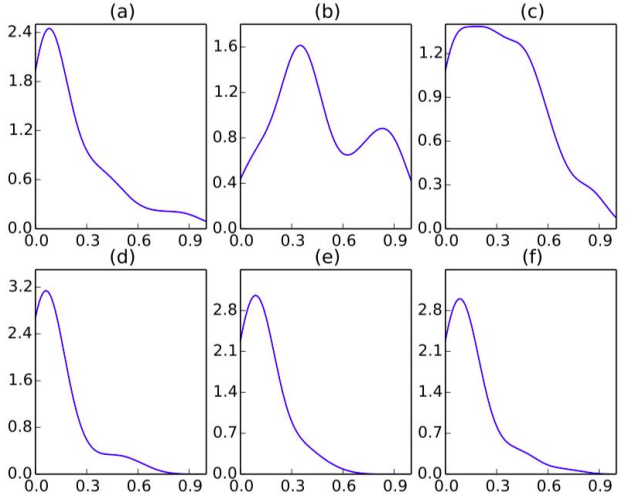
Experiments

Attention weights distribution

Attention weight distribution of “good”

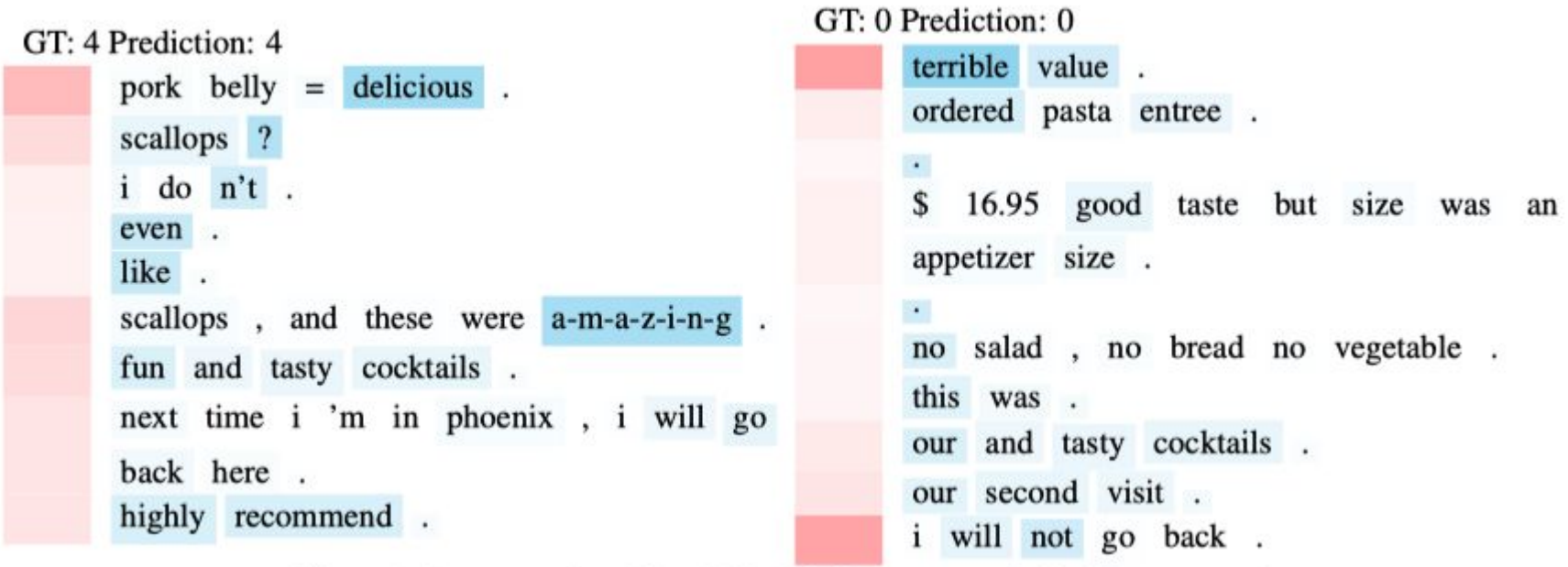


Attention weight distribution of “bad”



Experiments

Visualising attention



Conclusion

Further improvements

1. Use pre-trained contextual embeddings and skip BiGRU part on word level
2. Get word vectors directly from characters
3. Concatenate HAN embeddings with paragraph vector to improve classification

Conclusion

Possible applications

1. Multilingual attention networks https://arxiv.org/abs/1707.00896
2. Hierarchical attention networks for information extraction from cancer pathology reports

Conclusion

Final thoughts

1. Proposed intuitive and straightforward approach to build the document embeddings
2. Could be extended and modified for various tasks

Neural Discourse Structure for Text Categorization

—

Gaurav Kudva (UvA ID : 12205583)

Presentation Overview

1. Background Information
2. Model
3. Model Variants
4. Implementation Details
5. Datasets
6. Results
7. Qualitative Analysis
8. Summary
9. My Opinion

Background Information

Text Categorization

More heartening still, the climactic episode, “Avengers: Endgame,” succeeds at its daunting task: summing up an epic struggle with bedazzling action

Positive

Why do we need discourse structure

- Provides cues for the importance of different parts of a text
- Provides some sort of an inductive bias to models that incorporate the same

Rhetorical Structure Theory (RST)

- Document can be represented as a tree
  - Leaves are elementary discourse units (EDUs)
    - Can be sentences or clauses
  - Internal nodes represent the discourse relations across sentence spans
    - Some examples of discourse relations can be CONTRAST or ELABORATION
    - Sentence span
      - Nucleus - The more essential component of the span
      - Satellite - The supporting component of the span

Example of RST Tree

“Although the food was amazing and I was in love with the spicy pork burrito, the service was really awful. We watched our waiter serve himself many drinks. He kept running into the bathroom instead of grabbing our bill.”

A - [Although the food was amazing]

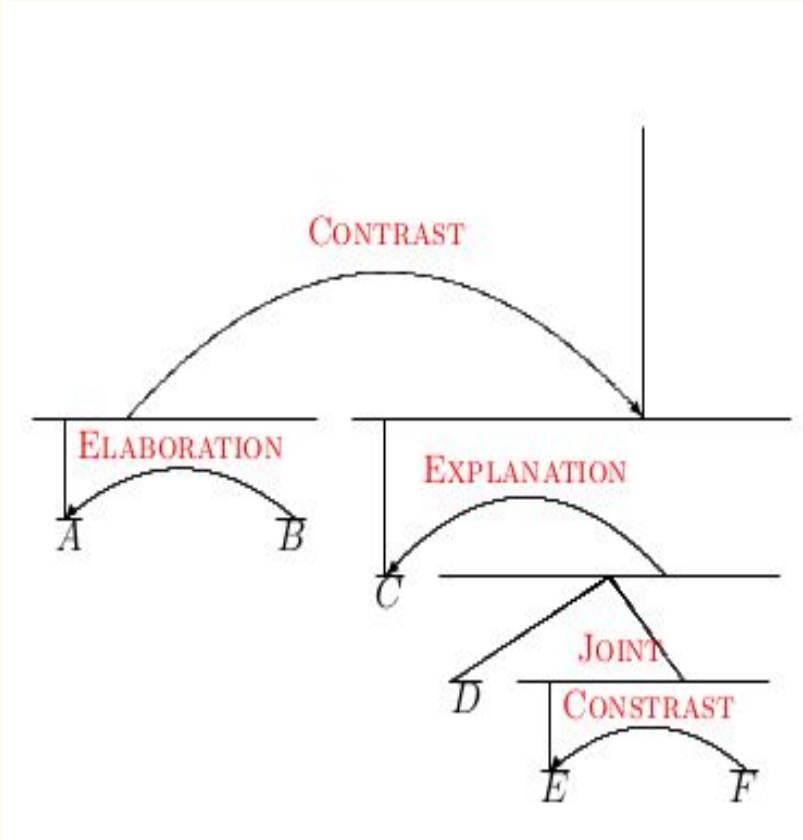
B - [and I was in love with the spicy pork burrito,]

C - [the service was really awful.]

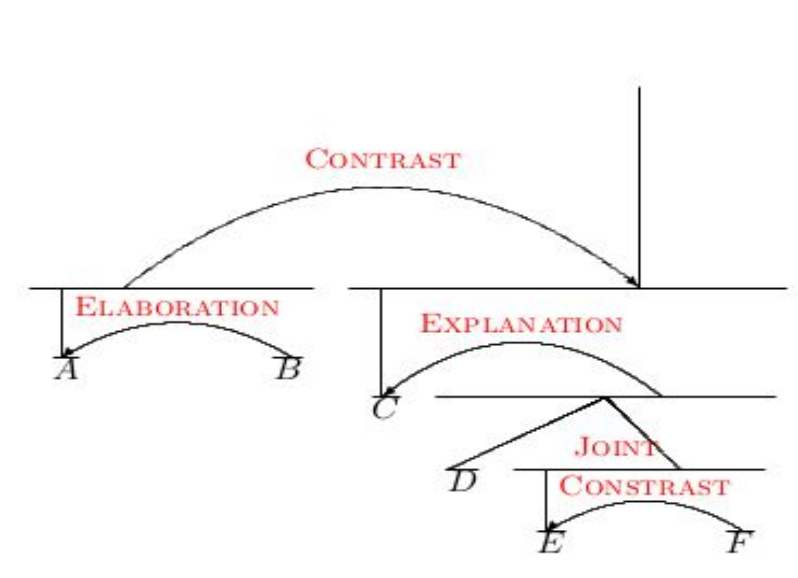
D - [We watched our waiter serve himself many drinks.]

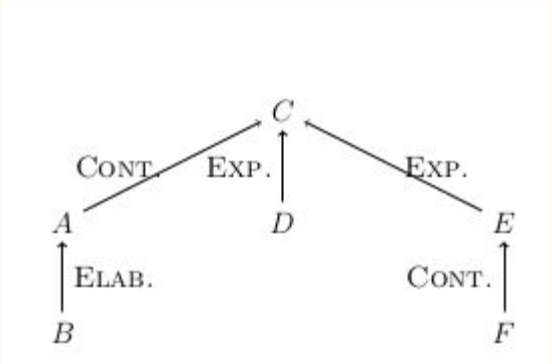
E - [He kept running into the bathroom]

F - [instead of grabbing our bill.]



RST Tree to Dependency Structure

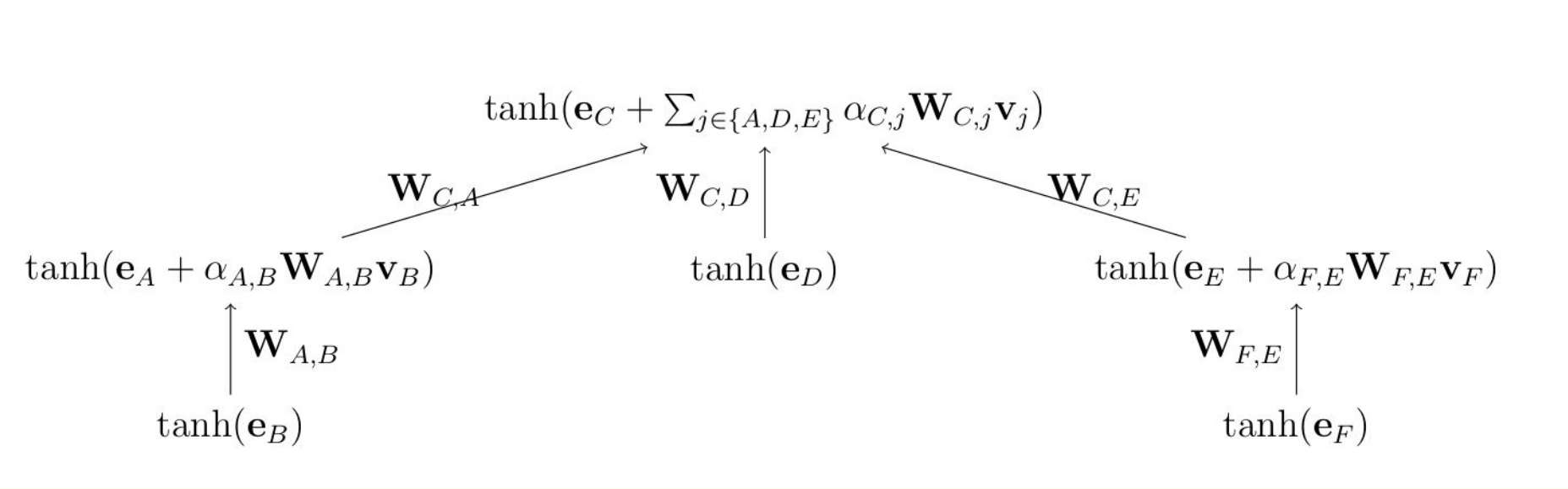




Most salient sentence : C - [“The service was really awful.”]

Model

Model Structure



Distributed Sentence Representation (e)

- Obtained through a **Bidirectional LSTM**
  - Concatenating the last hidden state of the forward and backward LSTM
    - $\mathbf{e} = [\overrightarrow{\mathbf{e}}; \overleftarrow{\mathbf{e}}]$

Full Recursive Model

- Builds a vector representation $\mathbf{v}_i$ for each node (i) in the tree
- If node (i) is

A leaf node

$$\mathbf{v}_i = \tanh(\mathbf{e}_i)$$

An internal node

$$\mathbf{v}_i = \tanh\left(\mathbf{e}_i + \sum_{j \in children(i)} \alpha_{i,j} \mathbf{W}_{r_{i,j}} \mathbf{v}_j\right)$$

Composition Function

$$\mathbf{v}_i = \tanh\left(\mathbf{e}_i + \sum_{j \in children(i)} \alpha_{i,j} \mathbf{W}_{r_{i,j}} \mathbf{v}_j\right)$$

- ★ - **Relation specific** composition matrix indexed by the relation between i and j
- ✦ - **Attention** weight

Attention Weights

$$\alpha_{i,j} = \sigma\left(\mathbf{e}_i^\top \mathbf{W}_\alpha \mathbf{v}_j\right)$$

- They are **not normalized**
  - Motivated by RST
- $\mathbf{W}_\alpha$ - Relation independent attention parameters

Category Prediction

$$\text{softmax}(\mathbf{W}_o \mathbf{v}_{\text{root}} + \mathbf{b})$$

Model Variants

Model Variants

- FULL model (the one we just discussed about)
- UNLABELED model
- ROOT model
- ADDITIVE model

UNLABELED Model

- To observe how absence of the relation labels affects the performance
- No relation-specific weight composition matrices
  - Reduced number of parameters
- Still uses the dependency tree to bias the model towards an approximation
- Composition Function

$$\mathbf{v}_i = \tanh\left(\mathbf{e}_i + \sum_{j \in children(i)} \alpha_{i,j}\mathbf{v}_j\right).$$

ROOT Model

- Uses the dependency structure to select the root EDU
  - $\mathbf{v}_{\text{root}} = \mathbf{e}_{\text{root}}$
- No composition function
- Based on the idea
  - Most central EDU is used to represent the whole document

ADDITIVE Model

- Does not utilize the dependency tree structure
- Simple composition function
  - $\mathbf{v}_{root}$ = Average of all the distributed embeddings ($\mathbf{e}_i$)

Implementation Details

Preprocessing

- Lowercase all tokens
- Remove tokens that contain only punctuation symbols
- Replace numbers with a special number token
- Low-frequency word types replaced by UNK
  - Reduce vocabulary for each dataset until 5% tokens are mapped to UNK

Discourse Parsing

- DPLP RST parser
  - Trained on 347 WSJ articles from the Penn Treebank
- RST trees are converted to dependency structures
  - Using methodology as described in Yoshida et al. (2014)

Word Embeddings

- Pre-trained GloVe embeddings
  - In the case of 10000 or fewer training examples
- Randomly initialized word embeddings
  - In the case of larger datasets
  - Trained alongside other parameters as well

Hyper-parameters

- Gradient norm clipping with threshold of 5
- Dropout of 0.3 on both input and hidden layers
- Grid search over
  - LSTM hidden state dimensionality [32, 48, 64, 128, 256]
  - Initial learning rate [0.1, 0.01, 0.001]
  - Optimizer [SGD, Adam]
- Highest-accuracy combination is selected
  - Using validation data or 10-fold cross validation

Datasets

Yelp Review Dataset

- Original dataset - 1.5 million examples
- Preprocessed dataset
  - 650,000 training and 50,000 test examples
- Task
  - Predict an ordinal rating (1-5) from the text of the review
- To select best combination of hyper-parameters
  - Randomly sample 10% of the training examples as validation data

Media Frames Corpus (MFC)

- 4,200 news articles on immigration from 13 U.S. newspapers (1980-2012)
- 15 general-purpose labels such as MORALITY, ECONOMICS
  - Focus on predicting the primary frame
- To select best hyper-parameter combination
  - Small set of examples as the validation set
  - Report average accuracy across 10-fold cross validation

Congressional Floor Debates Corpus

- Task
  - Predict “yea” or “nay” for the speaker of each speech segment
- Used the data split as suggested by Yessenalina et al. (2010)

Movie Review Corpus

- 1000 positive and 1000 negative reviews
- To select the best hyper-parameter combination
  - Average accuracy across folds using 10-fold cross validation

Congressional Bills Corpus

- 51,762 legislative bills from the 103rd to 111th U.S. Congresses
- Task
  - Whether a bill will survive based on its content
- To select the best hyper-parameter combination
  - Randomly sample 10% training samples as validation data

Results

Results

Method	Yelp	MFC	Debates	Movies	Bills
Prior work					
1. Yang et al. (2016)	71.0	—	—	—	—
2. Card et al. (2016)	—	56.8	—	—	—
3. Yogatama and Smith (2014)	—	—	74.0	—	88.5
4. Bhatia et al. (2015)	—	—	—	82.9	—
5. Hogenboom et al. (2015)	—	—	—	71.9	—
Variants of our model					
6. ADDITIVE	68.5	**57.6**	69.0	82.7	80.1
7. ROOT	54.3	51.2	60.3	68.7	70.5
8. UNLABELED	**71.3**	**58.4**	**75.7**	**83.1**	78.4
9. FULL	**71.8**	56.3	**74.2**	79.5	77.0

Relevant Observations

- Demonstrates benefit of using an explicit discourse structure
  - Even though an imperfect parser (trained on news text) has been used
  - Benefits vary based on the genre and different corpus sizes
- Even though the Congressional Bills Corpus has a large amount of data
  - The drop in accuracy is due to the high dissimilarity with
    - The data on which the RST parser is trained on

Qualitative Analysis

From DPLP:

$$1A \xrightarrow[\text{ELABORATION}]{0.66} 1B \xrightarrow[\text{CAUSE}]{0.67} 1C$$

[This store is somewhat convenient but I can never find any workers,]^{1A} [it drives me crazy.]^{1B} [I never come here on the weekends or around holidays anymore.]^{1C}

(a) true label: 2, predicted label: 2

From DPLP:

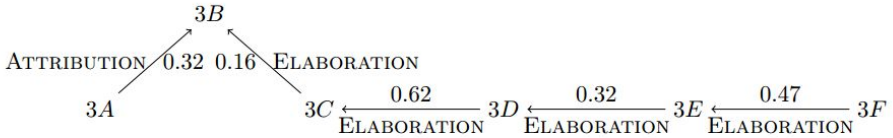
$$2C \xrightarrow[\text{EVALUATION}]{0.70} 2B \xrightarrow[\text{EVALUATION}]{0.87} 2A \xleftarrow[\text{ELABORATION}]{0.61} 2D$$

[I love these people.]^{2A} [They are very friendly and always ask about my life.]^{2B} [They remember things I tell them then ask about it the next time I'm in.]^{2C} [Patrick and Lily are the best but everyone there is wonderful in their own ways.]^{2D}

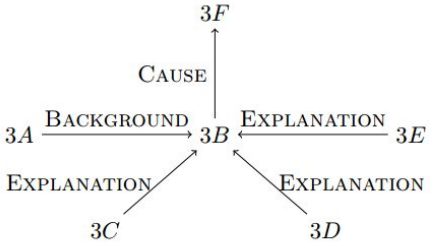
(b) true label: 5, predicted label: 5

Qualitative Analysis

From DPLP:



Manually constructed:

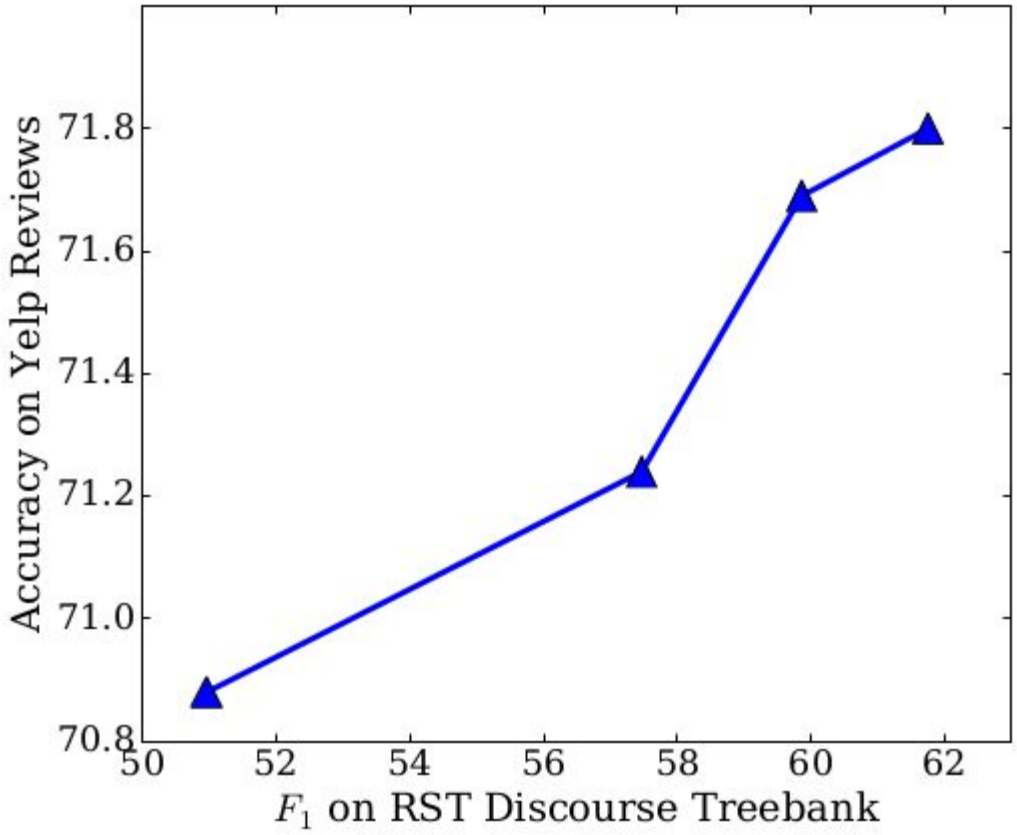


[We use to visit this pub 10 years ago because they had a nice english waitress and excellent fish and chips for the price.][3A] [However we went back a few weeks ago and were disappointed.][3B] [The price of the fish and chip dinner went up and they cut the portion in half.][3C] [No one assisted us in putting two tables together we had to do it ourselves.][3D] [Two guests wanted a good English hot tea and they didn't brew it in advance.][3E] [So we've decided there are newer and better places to eat fish and chips especially up in north phoenix.][3F]

(c) true label: 1, predicted label: 3

Effect of parsing performance

- Trained the RST parser on
  - 25%, 50% and 75% of the WSJ training set (random selection)
- Used the FULL model to predict the reviews in the Yelp Review dataset


Accuracy on Yelp Reviews
71.8
71.6
71.4
71.2
71.0
70.8
50
52
54
56
58
60
62
F_1 on RST Discourse Treebank

Attention Mechanism

Yelp Dataset

Normalized

$$\boldsymbol{\alpha}'_i = \text{softmax}\left(\begin{bmatrix} \vdots \\ \mathbf{v}_j^\top \\ \vdots \end{bmatrix}_{j \in children(i)} \mathbf{W}_\alpha \cdot \mathbf{e}_i\right)$$

70.3%

Un-normalized

$$\alpha_{i,j} = \sigma\left(\mathbf{e}_i^\top \mathbf{W}_\alpha \mathbf{v}_j\right)$$

71.8%

FULL Model

Final Takeaway

- Benefits provided by explicit discourse structure largely depend on
  - The quality of training the RST parser
  - The domain mismatch between the training corpus for a discourse parser and the domain where the discourse parser is used

My Thoughts

- Pros
  - Information is clear and concise
  - Good analysis on the theoretically driven deviation from the convention
    - Un-normalized attention
- Cons
  - To maintain the streak so far
    - No statistical significance
  - Not much information on the structure of the recursive neural network
- Future work
  - Experiment with different models to obtain the distributed representations
  - Domain adaptation methods to compensate for the domain mismatch

Thank You!