Learning Structured Text Representations

Yang Liu and **Mirella Lapata**
Institute for Language, Cognition and Computation
School of Informatics, University of Edinburgh
10 Crichton Street, Edinburgh EH8 9AB
`yang.liu2@ed.ac.uk,mlap@inf.ed.ac.uk`

Abstract

In this paper, we focus on learning structure-aware document representations from data without recourse to a discourse parser or additional annotations. Drawing inspiration from recent efforts to empower neural networks with a structural bias (Cheng et al., 2016; Kim et al., 2017), we propose a model that can encode a document while automatically inducing rich structural dependencies. Specifically, we embed a differentiable non-projective parsing algorithm into a neural model and use attention mechanisms to incorporate the structural biases. Experimental evaluations across different tasks and datasets show that the proposed model achieves state-of-the-art results on document modeling tasks while inducing intermediate structures which are both interpretable and meaningful.

1 Introduction

Document modeling is a fundamental task in Natural Language Processing useful to various downstream applications including topic labeling (Xie and Xing, 2013), summarization (Chen et al., 2016; Wolf and Gibson, 2006), sentiment analysis (Bhatia et al., 2015), question answering (Verberne et al., 2007), and machine translation (Meyer and Webber, 2013).

Recent work provides strong evidence that better document representations can be obtained by incorporating structural knowledge (Bhatia et al., 2015; Ji and Smith, 2017; Yang et al., 2016). Inspired by existing theories of discourse, representations of document structure have assumed several guises in the literature, such as trees in the style of Rhetorical Struc-

ture Theory (RST; Mann and Thompson, 1988), graphs (Lin et al., 2011; Wolf and Gibson, 2006), entity transitions (Barzilay and Lapata, 2008), or combinations thereof (Lin et al., 2011; Mesgar and Strube, 2015). The availability of discourse annotated corpora (Carlson et al., 2001; Prasad et al., 2008) has led to the development of off-the-shelf discourse parsers (e.g., Feng and Hirst, 2012; Liu and Lapata, 2017), and the common use of trees as representations of document structure. For example, Bhatia et al. (2015) improve document-level sentiment analysis by reweighing discourse units based on the depth of RST trees, whereas Ji and Smith (2017) show that a recursive neural network built on the output of an RST parser benefits text categorization in learning representations that focus on salient content.

Linguistically motivated representations of document structure rely on the availability of annotated corpora as well as a wider range of standard NLP tools (e.g., tokenizers, pos-taggers, syntactic parsers). Unfortunately, the reliance on labeled data, which is both difficult and highly expensive to produce, presents a major obstacle to the widespread use of discourse structure for document modeling. Moreover, despite recent advances in discourse processing, the use of an external parser often leads to pipeline-style architectures where errors propagate to later processing stages, affecting model performance.

It is therefore not surprising that there have been attempts to induce document representations directly from data without recourse to a discourse parser or additional annotations. The main idea is

63

to obtain *hierarchical* representations by first building representations of sentences, and then aggregating those into a document representation (Tang et al., 2015a,b). Yang et al. (2016) further demonstrate how to implicitly inject structural knowledge onto the representation using an attention mechanism (Bahdanau et al., 2015) which acknowledges that sentences are differentially important in different contexts. Their model learns to pay more or less attention to individual sentences when constructing the representation of the document.

Our work focus on learning deeper structure-aware document representations, drawing inspiration from recent efforts to empower neural networks with a structural bias (Cheng et al., 2016). Kim et al. (2017) introduce structured attention networks which are generalizations of the basic attention procedure, allowing to learn sentential representations while attending to partial segmentations or subtrees. Specifically, they take into account the dependency structure of a sentence by viewing the attention mechanism as a graphical model over latent variables. They first calculate unnormalized pairwise attention scores for all tokens in a sentence and then use the inside-outside algorithm to normalize the scores with the marginal probabilities of a dependency tree. Without recourse to an external parser, their model learns meaningful task-specific dependency structures, achieving competitive results in several sentence-level tasks. However, for document modeling, this approach has two drawbacks. Firstly, it does not consider non-projective dependency structures, which are common in document-level discourse analysis (Hayashi et al., 2016; Lee et al., 2006). As illustrated in Figure 1, the tree structure of a document can be flexible and the dependency edges may cross. Secondly, the inside-outside algorithm involves a dynamic programming process which is difficult to parallelize, making it impractical for modeling long documents.[1]

In this paper, we propose a new model for representing documents while automatically learning richer structural dependencies. Using a variant of Kirchhoff's Matrix-Tree Theorem (Tutte, 1984), our model implicitly considers non-projective depen-

---

[1]In our experiments, adding the inside-outside pass increases training time by a factor of 10.

1 The next time you hear a Member of Congress moan about the deficit, consider what Congress did Friday.
2 The Senate, 84-6 ,voted to increase to $124,000 the ceiling on insured mortgages from the FHA, which lost $4.2 billion in loan defaults last year.
3 Then , by voice vote , the Senate voted a porkbarrel bill, approved Thursday by the House, for domestic military construction.
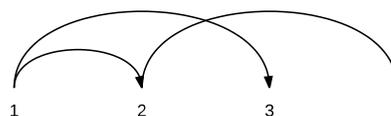4 The Bush request to what the Senators gave themselves.

Figure 1: The document is analyzed in the style of Rhetorical Structure Theory (Mann and Thompson, 1988), and represented as a dependency tree following the conversion algorithm of Hayashi et al. (2016).

dency tree structures. We keep each step of the learning process differentiable, so the model can be trained in an end-to-end fashion and induce discourse information that is helpful to specific tasks without an external parser. The inside-outside model of Kim et al. (2017) and our model both have a $O(n^3)$ worst case complexity. However, major operations in our approach can be parallelized efficiently on GPU computing hardware. Although our primary focus is on document modeling, there is nothing inherent in our model that prevents its application to individual sentences. Advantageously, it can induce non-projective structures which are required for representing languages with free or flexible word order (McDonald and Satta, 2007).

Our contributions in this work are threefold: a model for learning document representations whilst taking structural information into account; an efficient training procedure which allows to compute document level representations of arbitrary length; and a large scale evaluation study showing that the proposed model performs competitively against strong baselines while inducing intermediate structures which are both interpretable and meaningful.

2 Background

In this section, we describe how previous work uses the attention mechanism for representing individual sentences. The key idea is to capture the interaction between tokens within a sentence, generating a context representation for each word with weak structural information. This type of *intra-sentence* attention encodes relationships between words within
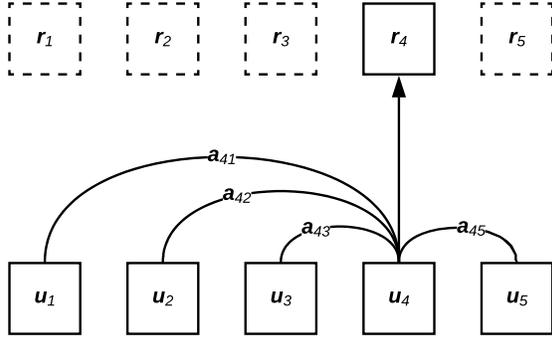
Figure 2: Intra-sentential attention mechanism; $\boldsymbol{a}_{ij}$ denotes the normalized attention score between tokens $\boldsymbol{u}_i$ and $\boldsymbol{u}_j$.

each sentence and differs from *inter-sentence* attention which has been widely applied to sequence transduction tasks like machine translation (Bahdanau et al., 2015) and learns the latent alignment *between* source and target sequences.

Figure 2 provides a schematic view of the intra-sentential attention mechanism. Given a sentence represented as a sequence of n word vectors $[\boldsymbol{u}_1, \boldsymbol{u}_2, \cdots, \boldsymbol{u}_n]$, for each word pair $\langle \boldsymbol{u}_i, \boldsymbol{u}_j \rangle$, the attention score $\boldsymbol{a}_{ij}$ is estimated as:

$$\boldsymbol{f}_{ij} = F(\boldsymbol{u}_i, \boldsymbol{u}_j) \qquad (1)$$

$$\boldsymbol{a}_{ij} = \frac{exp(\boldsymbol{f}_{ij})}{\sum_{k=1}^{n} exp(\boldsymbol{f}_{ik})} \qquad (2)$$

where $F()$ is a function for computing the unnormalized score $\boldsymbol{f}_{ij}$ which is then normalized by calculating a probability distribution $\boldsymbol{a}_{ij}$. Individual words collect information from their context based on $\boldsymbol{a}_{ij}$ and obtain a context representation:

$$\boldsymbol{r}_i = \sum_{j=1}^{n} \boldsymbol{a}_{ij} \boldsymbol{u}_j \qquad (3)$$

where attention score $\boldsymbol{a}_{ij}$ indicates the (dependency) relation between the i-th and the j-th-words and how information from $\boldsymbol{u}_j$ should be fed into $\boldsymbol{u}_i$.

Despite successful application of the above attention mechanism in sentiment analysis (Cheng et al., 2016) and entailment recognition (Parikh et al., 2016), the structural information under consideration is shallow, limited to word-word dependencies. Since attention is computed as a simple probability distribution, it cannot capture more elaborate

structural dependencies such as trees (or graphs). Kim et al. (2017) induce richer internal structure by imposing structural constraints on the probability distribution computed by the attention mechanism. Specifically, they normalize $\boldsymbol{f}_{ij}$ with a projective dependency tree using the inside-outside algorithm (Baker, 1979):

$$\boldsymbol{f}_{ij} = F(\boldsymbol{u}_i, \boldsymbol{u}_j) \qquad (4)$$

$$\boldsymbol{a} = inside\text{-}outside(\boldsymbol{f}) \qquad (5)$$

$$\boldsymbol{r}_i = \sum_{j=1}^{n} \boldsymbol{a}_{ij} \boldsymbol{u}_j \qquad (6)$$

This process is differentiable, so the model can be trained end-to-end and learn structural information without relying on a parser. However, efficiency is a major issue, since the inside-outside algorithm has time complexity $O(n^3)$ (where n represents the number of tokens) and does not lend itself to easy parallelization. The high order complexity renders the approach impractical for real-world applications.

3 Encoding Text Representations

In this section we present our document representation model. We follow previous work (Tang et al., 2015a; Yang et al., 2016) in modeling documents *hierarchically* by first obtaining representations for sentences and then composing those into a document representation. Structural information is taken into account while learning representations for both sentences and documents and an attention mechanism is applied on both words within a sentence and sentences within a document. The general idea is to force pair-wise attention between text units to form a non-projective dependency tree, and automatically induce this tree for different natural language processing tasks in a differentiable way. In the following, we first describe how the attention mechanism is applied to sentences, and then move on to present our document-level model.

3.1 Sentence Model

Let $T = [\boldsymbol{u}_1, \boldsymbol{u}_2, \cdots, \boldsymbol{u}_n]$ denote a sentence containing a sequence of words, each represented by a vector $\boldsymbol{u}$, which can be pre-trained on a large corpus. Long Short-Term Memory Neural Networks (LSTMs; Hochreiter and Schmidhuber, 1997) have
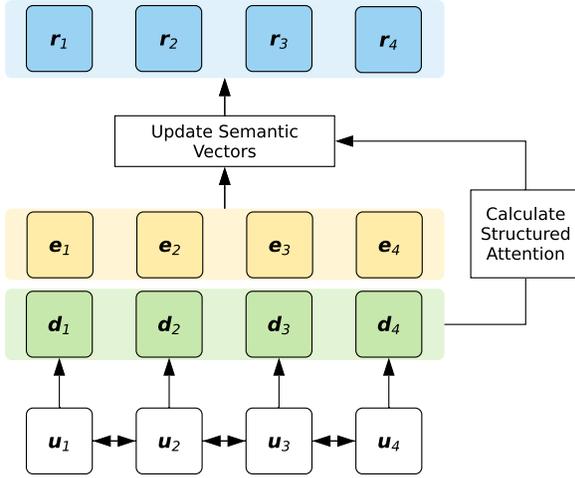
Figure 3: Sentence representation model: $\boldsymbol{u}_t$ is the input vector for the t-th word, $\boldsymbol{e}_t$ and $\boldsymbol{d}_t$ are semantic and structure vectors, respectively.

been successfully applied to various sequence modeling tasks ranging from machine translation (Bahdanau et al., 2015), to speech recognition (Graves et al., 2013), and image caption generation (Xu et al., 2015). In this paper we use bidirectional LSTMs as a way of representing elements in a sequence (i.e., words or sentences) together with their contexts, capturing the element and an "infinite" window around it. Specifically, we run a bidirectional LSTM over sentence T, and take the output vectors $[\boldsymbol{h}_1, \boldsymbol{h}_2, \cdots, \boldsymbol{h}_n]$ as the representations of words in T, where $\boldsymbol{h}_t \in \mathbb{R}^k$ is the output vector for word $\boldsymbol{u}_t$ based on its context.

We then exploit the structure of T which we induce based on an attention mechanism detailed below to obtain more precise representations. Inspired by recent work (Daniluk et al., 2017; Miller et al., 2016), which shows that the conventional way of using LSTM output vectors for calculating both attention and encoding word semantics is overloaded and likely to cause performance deficiencies, we decompose the LSTM output vector in two parts:

$$[\boldsymbol{e}_t, \boldsymbol{d}_t] = \boldsymbol{h}_t \qquad (7)$$

where $\boldsymbol{e}_t \in \mathbb{R}^{k_t}$, the semantic vector, encodes semantic information for specific tasks, and $\boldsymbol{d}_t \in \mathbb{R}^{k_s}$, the structure vector, is used to calculate structured attention.

We use a series of operations based on the Matrix-Tree Theorem (Tutte, 1984) to incorporate the struc-

tural bias of non-projective dependency trees into the attention weights. We constrain the probability distributions $\boldsymbol{a}_{ij}$ (see Equation (2)) to be the posterior marginals of a dependency tree structure. We then use the normalized structured attention, to build a context vector for updating the semantic vector of each word, obtaining new representations $[\boldsymbol{r}_1, \boldsymbol{r}_2, \cdots, \boldsymbol{r}_n]$. An overview of the model is presented in Figure 3. We describe the attention mechanism in detail in the following section.

3.2 Structured Attention Mechanism

Dependency representations of natural language are a simple yet flexible mechanism for encoding words and their syntactic relations through directed graphs. Much work in descriptive linguistics (Melčuk, 1988; Tesniére, 1959) has advocated their suitability for representing syntactic structure across languages. A primary advantage of dependency representations is that they have a natural mechanism for representing discontinuous constructions arising from long distance dependencies or free word order through non-projective dependency edges.

More formally, building a dependency tree amounts to finding latent variables z_{ij} for all $i \neq j$, where word i is the parent node of word j, under some global constraints, amongst which the single-head constraint is the most important, since it forces the structure to be a rooted tree. We use a variant of Kirchhoff's Matrix-Tree Theorem (Koo et al., 2007; Tutte, 1984) to calculate the marginal probability of each dependency edge $P(z_{ij} = 1)$ of a non-projective dependency tree, and this probability is used as the attention weight that decides how much information is collected from child unit j to the parent unit i.

We first calculate unnormalized attention scores $\boldsymbol{f}_{ij}$ with structure vector $\boldsymbol{d}$ (see Equation (7)) via a bilinear function:

$$\boldsymbol{t}_p = tanh(\boldsymbol{W}_p \boldsymbol{d}_i) \qquad (8)$$
$$\boldsymbol{t}_c = tanh(\boldsymbol{W}_c \boldsymbol{d}_j) \qquad (9)$$
$$\boldsymbol{f}_{ij} = \boldsymbol{t}_p^T \boldsymbol{W}_a \boldsymbol{t}_c \qquad (10)$$

where $\boldsymbol{W}_p \in \mathbb{R}^{k_s * k_s}$ and $\boldsymbol{W}_c \in \mathbb{R}^{k_s * k_s}$ are the weights for building the representation of parent and child nodes. $\boldsymbol{W}_a \in \mathbb{R}^{k_s * k_s}$ is the weight for the bilinear transformation. $\boldsymbol{f} \in \mathbb{R}^{n * n}$ can be viewed as

a weighted adjacency matrix for a graph G with n nodes where each node corresponds to a word in a sentence. We also calculate the root score $\boldsymbol{f}_i^r$, indicating the unnormalized possibility of a node being the root:

$$\boldsymbol{f}_i^r = \boldsymbol{W}_r \boldsymbol{d}_i \qquad (11)$$

where $\boldsymbol{W}_r \in \mathbb{R}^{1*k_s}$. We calculate $P(z_{ij} = 1)$, the marginal probability of the dependency edge, following Koo et al. (2007):

$$\boldsymbol{A}_{ij} = \begin{cases} 0 & \text{if } i = j \\ exp(\boldsymbol{f}_{ij}) & \text{otherwise} \end{cases} \qquad (12)$$

$$\boldsymbol{L}_{ij} = \begin{cases} \sum_{i'=1}^{n} \boldsymbol{A}_{i'j} & \text{if } i = j \\ -\boldsymbol{A}_{ij} & \text{otherwise} \end{cases} \qquad (13)$$

$$\bar{\boldsymbol{L}}_{ij} = \begin{cases} exp(\boldsymbol{f}_i^r) & i = 1 \\ \boldsymbol{L}_{ij} & i > 1 \end{cases} \qquad (14)$$

$$P(z_{ij} = 1) = (1 - \delta_{1,j}) \boldsymbol{A}_{ij} [\bar{\boldsymbol{L}}^{-1}]_{jj} \\ - (1 - \delta_{i,1}) \boldsymbol{A}_{ij} [\bar{\boldsymbol{L}}^{-1}]_{ji} \qquad (15)$$

$$P(root(i)) = exp(f_r^i) [\bar{\boldsymbol{L}}^{-1}]_{i1}$$

where $1 \leq i \leq n, 1 \leq j \leq n$. $\boldsymbol{L} \in \mathbb{R}^{n*n}$ is the Laplacian matrix for graph G and $\bar{\boldsymbol{L}} \in \mathbb{R}^{n*n}$ is a variant of $\boldsymbol{L}$ that takes the root node into consideration, and δ is the Kronecker delta. The key for the calculation to hold is for $\boldsymbol{L}^{ii}$, the minor of the Laplacian matrix $\boldsymbol{L}$ with respect to row i and column i, to be equal to the sum of the weights of all directed spanning trees of G which are rooted at i. $P(z_{ij} = 1)$ is the marginal probability of the dependency edge between the i-th and j-th words. $P(root(i) = 1)$ is the marginal probability of the i-th word headed by the root of the tree. Details of the proof can be found in Koo et al. (2007).

We denote the marginal probabilities $P(z_{ij} = 1)$ as $\boldsymbol{a}_{ij}$ and $P(root(i))$ as $\boldsymbol{a}_i^r$. This can be interpreted as attention scores which are constrained to converge to a structured object, a non-projective dependency tree, in our case. We update the semantic vector $\boldsymbol{e}_i$ of each word with structured attention:

$$\boldsymbol{p}_i = \sum_{k=1}^{n} \boldsymbol{a}_{ki} \boldsymbol{e}_k + \boldsymbol{a}_i^r \boldsymbol{e}_{root} \qquad (16)$$

$$\boldsymbol{c}_i = \sum_{k=1}^{n} \boldsymbol{a}_{ik} \boldsymbol{e}_i \qquad (17)$$

$$\boldsymbol{r}_i = tanh(\boldsymbol{W}_r [\boldsymbol{e}_i, \boldsymbol{p}_i, \boldsymbol{c}_i]) \qquad (18)$$

where $\boldsymbol{p}_i \in \mathbb{R}^{k_e}$ is the context vector gathered from possible parents of u_i and $\boldsymbol{c}_i \in \mathbb{R}^{k_e}$ the context vector gathered from possible children, and $\boldsymbol{e}_{root}$ is a special embedding for the root node. The context vectors are concatenated with $\boldsymbol{e}_i$ and transformed with weights $\boldsymbol{W}_r \in \mathbb{R}^{k_e*3k_e}$ to obtain the updated semantic vector $\boldsymbol{r}_i \in \mathbb{R}^{k_e}$ with rich structural information (see Figure 3).

3.3 Document Model

We build document representations hierarchically: sentences are composed of words and documents are composed of sentences. Composition on the document level also makes use of structured attention in the form of a dependency graph. Dependency-based representations have been previously used for developing discourse parsers (Hayashi et al., 2016; Li et al., 2014) and in applications such as summarization (Hirao et al., 2013).

As illustrated in Figure 4, given a document with n sentences $[s_1, s_2, \cdots, s_n]$, for each sentence s_i, the input is a sequence of word embeddings $[\boldsymbol{u}_{i1}, \boldsymbol{u}_{i2}, \cdots, \boldsymbol{u}_{im}]$, where m is the number of tokens in s_i. By feeding the embeddings into a sentence-level bi-LSTM and applying the proposed structured attention mechanism, we obtain the updated semantic vector $[\boldsymbol{r}_{i1}, \boldsymbol{r}_{i2}, \cdots, \boldsymbol{r}_{im}]$. Then a pooling operation produces a fixed-length vector $\boldsymbol{v}_i$ for each sentence. Analogously, we view the document as a sequence of sentence vectors $[\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_n]$ whose embeddings are fed to a document-level bi-LSTM. Application of the structured attention mechanism creates new semantic vectors $[\boldsymbol{q}_1, \boldsymbol{q}_2, \cdots, \boldsymbol{q}_n]$ and another pooling operation yields the final document representation $\boldsymbol{y}$.

3.4 End-to-End Training

Our model can be trained in an end-to-end fashion since all operations required for computing structured attention and using it to update the semantic
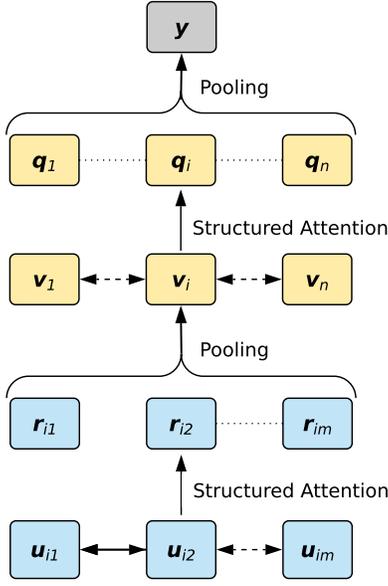
67

Figure 4: Document representation model.

vectors are differentiable. In contrast to in Kim et al. (2017), training can be done efficiently. The major complexity of our model lies in the computation of the gradients of the the inverse matrix. Let A denote a matrix depending on a real parameter x; assuming all component functions in A are differentiable, and A is invertible for all possible values, the gradient of A with respect respect to x is:

$$\frac{dA^{-1}}{dx} = -A^{-1}\frac{dA}{dx}A^{-1} \qquad (19)$$

Multiplication of the three matrices and matrix inversion can be computed efficiently on modern parallel hardware architectures such as GPUs. In our experiments, computation of structured attention takes only 1/10 of training time.

4 Experiments

In this section we present our experiments for evaluating the performance of our model. Since sentence representations constitute the basic building blocks of our document model, we first evaluate the performance of structured attention on a sentence-level task, namely natural language inference. We then assess the document-level representations obtained by our model on a variety of classification tasks representing documents of different length, subject matter, and language. Our

code is available at `https://github.com/nlpyang/structured`.

4.1 Natural Language Inference

The ability to reason about the semantic relationship between two sentences is an integral part of text understanding. We therefore evaluate our model on recognizing textual entailment, i.e., whether two premise-hypothesis pairs are entailing, contradictory, or neutral. For this task we used the Stanford Natural Language Inference (SNLI) dataset (Bowman et al., 2015), which contains premise-hypothesis pairs and target labels indicating their relation. After removing sentences with unknown labels, we obtained 549,367 pairs for training, 9,842 for development and 9,824 for testing.

Sentence-level representations obtained by our model (with structured attention) were used to encode the premise and hypothesis by modifying the model of Parikh et al. (2016) as follows. Let $[\boldsymbol{x}_1^p, \cdots, \boldsymbol{x}_n^p]$ and $[\boldsymbol{x}_1^h, \cdots, \boldsymbol{x}_m^h]$ be the input vectors for the premise and hypothesis, respectively. Application of structured attention yields new vector representations $[\boldsymbol{r}_1^p, \cdots, \boldsymbol{r}_n^p]$ and $[\boldsymbol{r}_1^h, \cdots, \boldsymbol{r}_m^h]$. Then we combine these two vectors with inter-sentential attention, and apply an average pooling operation:

$$\boldsymbol{o}_{ij} = MLP(\boldsymbol{r}_i^p)^T MLP(\boldsymbol{r}_j^h) \qquad (20)$$

$$\bar{\boldsymbol{r}}_i^p = [\boldsymbol{r}_i^p, \sum_{j=1}^m \frac{exp(\boldsymbol{o}_{ij})}{\sum_{k=1}^m exp(\boldsymbol{o}_{ik})}] \qquad (21)$$

$$\bar{\boldsymbol{r}}_i^h = [\boldsymbol{r}_i^h, \sum_{i=1}^m \frac{exp(\boldsymbol{o}_{ij})}{\sum_{k=1}^m exp(\boldsymbol{o}_{kj})}] \qquad (22)$$

$$\boldsymbol{r}^p = \sum_{i=1}^n g(\bar{\boldsymbol{r}}_i^p), \quad \boldsymbol{r}^h = \sum_{i=1}^m g(\bar{\boldsymbol{r}}_i^h) \qquad (23)$$

where $MLP()$ is a two-layer perceptron with a $ReLU$ activation function. The new representations $\boldsymbol{r}^p, \boldsymbol{r}^h$ are then concatenated and fed into another two-layer perceptron with a softmax layer to obtain the predicted distribution over the labels.

The hidden size of the LSTM was set to 150. The dimensions of the semantic vector were 100 and the dimensions of structure vector were 50. We used pretrained 300-D Glove 840B (Pennington et al., 2014) vectors to initialize the word embeddings. All parameters (including word embeddings) were updated with Adagrad (Duchi et al., 2011), and the

Models	Acc	θ
Classifier with handcrafted features (Bowman et al., 2015)	78.2	—
300D LSTM encoders (Bowman et al., 2015)	80.6	3.0M
300D Stack-Augmented Parser-Interpreter Neural Net (Bowman et al., 2016)	83.2	3.7M
100D LSTM with inter-attention (Rocktäschel et al., 2016)	83.5	252K
200D Matching LSTMs (Wang and Jiang, 2016)	86.1	1.9M
450D LSTMN with deep attention fusion (Cheng et al., 2016)	86.3	3.4M
Decomposable Attention over word embeddings (Parikh et al., 2016)	86.8	582K
Enhanced BiLSTM Inference Model (Chen et al., 2017)	**88.0**	4.3M
175D No Attention	85.3	600K
175D Simple intra-sentence attention	86.2	1.1M
100D Structured intra-sentence attention with Inside-Outside	86.8	1.2M
175D Structured intra-sentence attention with Matrix Inversion	**86.9**	1.1M

Table 1: Test accuracy on the SNLI dataset and number of parameters θ (excluding embeddings). Wherever available we also provide the size of the recurrent unit.

Models	Speed	
	Max	Avg
No Attention	0.0050	0.0033
Simple Attention	0.0057	0.0042
Matrix Inversion	0.0070	0.0045
Inside-Outside	0.1200	0.0380

Table 2: Comparison of speed of different models on the SNLI testset. The unit of measurement is seconds per instance. All results were obtained on a GeForce GTX TITAN X (Pascal) GPU.

Dataset	#class	#docs	#s/d	#w/d
Yelp	5	335K	8.9	151.6
IMDB	10	348K	14.0	325.6
CZ Movies	3	92K	3.5	51.2
Debates	2	1.6K	22.7	519.2

Table 3: Dataset statistics; #class is the number of classes per dataset, #docs denotes the number of documents; #s/d and #w/d represent the average number of sentences and words per document.

learning rate was set to 0.05. The hidden size of the two-layer perceptron was set to 200 and dropout was used with ratio 0.2. The mini-batch size was 32.

We compared our model (and variants thereof) against several related systems. Results (in terms of 3-class accuracy) are shown in Table 1. Most previous systems employ LSTMs and do not incorporate a structured attention component. Exceptions include Cheng et al. (2016) and Parikh et al. (2016) whose models include intra-attention encoding relationships between words within each sentence (see Equation (2)). It is also worth noting that some models take structural information into account in the form of parse trees (Bowman et al., 2016; Chen et al., 2017). The second block of Table 1 presents a version of our model without an intra-sentential attention mechanism as well as three variants with attention, assuming the structure of word-to-word re-

lations and dependency trees. In the latter case we compare our matrix inversion based model against Kim et al.'s (2017) inside-outside attention model. Consistent with previous work (Cheng et al., 2016; Parikh et al., 2016), we observe that simple attention brings performance improvements over no attention. Structured attention further enhances performance. Our own model with tree matrix inversion slightly outperforms the inside-outside model of Kim et al. (2017), overall achieving results in the same ballpark with related LSTM-based models (Chen et al., 2017; Cheng et al., 2016; Parikh et al., 2016).

Table 2 compares the running speed of the models shown in the second block of Table 1. As can be seen matrix inversion does not increase running speed over the simpler attention mechanism and is considerably faster compared to inside-outside. The latter is 10–20 times slower than our model on the same platform.

Models	Yelp	IMDB	CZ Movies	Debates	θ
Feature-based classifiers	59.8	40.9	78.5	74.0	—
Paragraph vector (Tang et al., 2015a)	57.7	34.1	—	—-	—
Convolutional neural network (Tang et al., 2015a)	59.7	—	—	—	—
Convolutional gated RNN (Tang et al., 2015a)	63.7	42.5	—	—	—
LSTM gated RNN (Tang et al., 2015a)	65.1	45.3	—	—	—
RST-based recursive neural network (Ji and Smith, 2017)	—	—	—	75.7	—
75D Hierarchical attention networks (Yang et al., 2016)	68.2	**49.4**	80.8	74.0	273K
75D No Attention	66.7	47.5	80.5	73.7	330K
100D Simple Attention	67.7	48.2	81.4	75.3	860K
100D Structured Attention (sentence-level)	68.0	48.8	81.5	74.6	842K
100D Structured Attention (document-level)	67.8	48.6	81.1	75.2	842K
100D Structured Attention (both levels)	**68.6**	49.2	**82.1**	**76.5**	860K

Table 4: Test accuracy on four datasets and number of parameters θ (excluding embeddings). Regarding feature-based classification methods, results on Yelp and IMDB are taken from Tang et al. (2015a), on CZ movies from Brychcın and Habernal (2013), and Debates from Yogatama and Smith (2014). Wherever available we also provide the size of the recurrent unit (LSTM or GRU).

4.2 Document Classification

In this section, we evaluate our document-level model on a variety of classification tasks. We selected four datasets which we describe below. Table 3 summarizes some statistics for each dataset.

Yelp reviews were obtained from the 2013 Yelp Dataset Challenge. This dataset contains restaurant reviews, each associated with human ratings on a scale from 1 (negative) to 5 (positive) which we used as gold labels for sentiment classification; we followed the preprocessing introduced in Tang et al. (2015a) and report experiments on their training, development, and testing partitions (80/10/10).

IMDB reviews were obtained from Diao et al. (2014), who randomly crawled reviews for 50K movies. Each review is associated with user ratings ranging from 1 to 10.

Czech reviews were obtained from Brychcın and Habernal (2013). The dataset contains reviews from the Czech Movie Database[2] each labeled as positive, neutral, or negative. We include Czech in our experiments since it has more flexible word order compared to English, with non-projective dependency structures being more frequent. Experiments on this dataset perform 10-fold cross-validation following previous work (Brychcın and Habernal, 2013).

Congressional floor debates were obtained from a corpus originally created by Thomas et al. (2006) which contains transcripts of U.S. floor debates in the House of Representatives for the year 2005. Each debate consists of a series of speech segments, each labeled by the vote ("yea" or "nay") cast for the proposed bill by the the speaker of each segment. We used the pre-processed corpus from Yogatama and Smith (2014).[3]

Following previous work (Yang et al., 2016), we only retained words appearing more than five times in building the vocabulary and replaced words with lesser frequencies with a special UNK token. Word embeddings were initialized by training word2vec (Mikolov et al., 2013) on the training and validation splits of each dataset. In our experiments, we set the word embedding dimension to be 200 and the hidden size for the sentence-level and document-level LSTMs to 100 (the dimensions of the semantic and structure vectors were set to 75 and 25, respectively). We used a mini-batch size of 32 during training and documents of similar length were grouped in one batch. Parameters were optimized with Adagrad (Duchi et al., 2011), the learning rate was set to 0.05. We used L_2 regularization for all parameters except word embeddings with regularization constant set to $1e^{-4}$. Dropout was applied on the input and output

---

[2]http://www.csfd.cz/

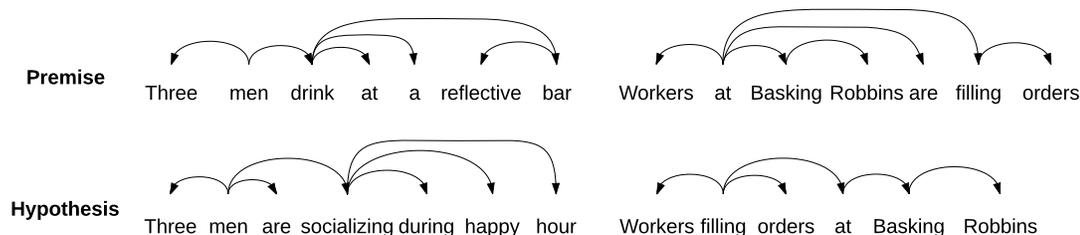[3]http://www.cs.cornell.edu/~ainur/data.html

Figure 5: Dependency trees induced by our model on the SNLI test set.

layers with dropout rate 0.3.

Our results are summarized in Table 4. We compared our model against several related models covering a wide spectrum of representations including word-based ones (e.g., paragraph vector and CNN models) as well as hierarchically composed ones (e.g., a CNN or LSTM provides a sentence vector and then a recurrent neural network combines the sentence vectors to form a document level representation for classification). Previous state-of-the-art results on the three review datasets were achieved by the hierarchical attention network of Yang et al. (2016), which models the document hierarchically with two GRUs and uses an attention mechanism to weigh the importance of each word and sentence. On the debates corpus, Ji and Smith (2017) obtained best results with a recursive neural network model operating on the output of an RST parser. Table 4 presents three variants[4] of our model, one with structured attention on the sentence level, another one with structured attention on the document level and a third model which employs attention on both levels. As can be seen, the combination is beneficial achieving best results on three out of four datasets. Furthermore, structured attention is superior to the simpler word-to-word attention mechanism, and both types of attention bring improvements over no attention. The structured attention approach is also very efficient, taking only 20 minutes for one training epoch on the largest dataset.

4.3 Analysis of Induced Structures

To gain further insight on structured attention, we inspected the dependency trees it produces. Specifically, we used the Chu-Liu-Edmonds algo-

---

[4]We do not report comparisons with the inside-outside approach on document classification tasks due to its prohibitive computation cost leading to 5 hours of training for one epoch.

		Parser	Attention
Projective		—	51.4%
Height		8.99	5.78
Nodes	depth 1	9.8%	8.4%
	depth 2	15.0%	19.7%
	depth 3	12.8%	22.4%
	depth 4	12.5%	23.4%
	depth 5	12.0%	14.4%
	depth 6	10.3%	4.5%
Same Edges		38.7%	

Table 5: Descriptive statistics for dependency trees produced by our model and the Stanford parser (Manning et al., 2014) on the SNLI test set.

rithm (Chu and Liu, 1965; Edmonds, 1967) to extract the maximum spanning tree from the attention scores. We report various statistics on the characteristics of the induced trees across different tasks and datasets. We also provide examples of tree output, in an attempt to explain how our model uses dependency structures to model text.

Sentence Trees We compared the dependency trees obtained from our model with those produced by a state-of-the-art dependency parser trained on the English Penn Treebank. Table 5 presents various statistics on the depth of the trees produced by our model on the SNLI test set and the Stanford dependency parser (Manning et al., 2014). As can be seen, the induced dependency structures are simpler compared to those obtained from the Stanford parser. The trees are generally less deep (their height is 5.78 compared to 8.99 for the Stanford parser), with the majority being of depth 2–4. Almost half of the induced trees have a projective structure, although there is nothing in the model to enforce this constraint. We also calculated the percentage of head-dependency edges that are identical between the two

|		Yelp	IMDB	CZ Movies	Debates
Projective		79.6%	74.9%	82.8%	62.4%
Height		2.81	3.34	1.50	3.58
Nodes	depth 2	15.1%	13.6%	25.7%	12.8%
	depth 3	55.6%	46.8%	57.1%	30.2%
	depth 4	22.3%	32.5%	11.3%	40.8%
	depth 5	3.2%	4.1%	5.8%	14.8%

Table 6: Descriptive statistics for induced document-level dependency trees across datasets.

sets of trees. Although our model is not exposed to annotated trees during training, a large number of edges agree with the output of the Stanford parser.

Figure 5 shows examples of dependency trees induced on the SNLI dataset. Although the model is trained without ever being exposed to a parse tree, it is able to learn plausible dependency structures via the attention mechanism. Overall we observe that the induced trees differ from linguistically motivated ones in the types of dependencies they create which tend to be of shorter length. The dependencies obtained from structured attention are more direct as shown in the first premise sentence in Figure 5 where words *at* and *bar* are directly connected to the verb *drink*. This is perhaps to be expected since the attention mechanism uses the dependency structures to collect information from other words, and the direct links will be more effective.

Document Trees We also used the Chu-Liu-Edmonds algorithms to obtain document-level dependency trees. Table 6 summarizes various characteristics of these trees. For most datasets, document-level trees are not very deep, they mostly contain up to nodes of depth 3. This is not surprising as the documents are relatively short (see Table 3) with the exception of debates which are longer and the induced trees more complex. The fact that most documents exhibit simple discourse structures is further corroborated by the large number (over 70%) of projective trees induced on Yelp, IMBD, and CZ Movies datasets. Unfortunately, our trees cannot be directly compared with the output of a discourse parser which typically involves a segmentation process splitting sentences into smaller units. Our trees are constructed over entire sentences, and there is no mechanism currently in the model to split sentences

1 first of all, i did not expect to come into a cafeteria style eatery.
2 they serve the basics of bbq, nothing too fancy.
3 a few appetizers and side options, like cheesy potatoes, baked mac 'n' 4 cheese, fresh corn bread, etc..
4 all were very tasty.
5 for entree, they have a wide variety of meats and combos and samplers.
6 overall, this is a great place,... meat was well prepared, a little pricey for what i was expecting.

(a)
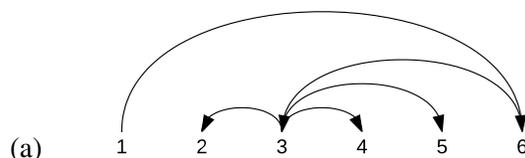
1 great instruction by ryan
2 clean workout facility and friendly people
3 they have a new student membership for 60 per month and classes are mon , weds and fri 6pm 7pm
4 it 's definitely worth money if you want to learn brazilian jiu jitsu
5 i usually go to classes on mondays and fridays , and it 's the best workout i 've had in years

(b)
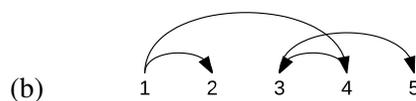
1 Udělat parodii tak, aby nebyla ještě trapnější než původní film, není zrovna legrace, o tom jsem se přesvědčila už několikrát (Bulšit, Scary Movie...).
   To make a parody so that it ends up being even more embarrassing than the original movie is not exactly trivial, this I have been convinced of several times already (Bullshit, Scary Movie...).
2 Jenže Top Secret?
   But Top Secret?
3 Nevím, jestli to vůbec můžu říct, ale možná tenhle film překonal i skvělé Žhavé výstřely!
   I don't know if I can actually say it, but maybe this movie has scored even better than the fantastic Hot Shots!
4 Bože, to jsem se nasmála!
   God, I laughed a lot!
5 Nutno uznat, že je to docela síla, když si Američané dělali před revolucí takovou srandu z Němců.
   I must admit, that it's pretty cool, when Americans were making so much fun of Germans before the revolution.
6 Ty narážky byly vážně skvělé... Jedna z nejlepších parodií, co jsem kdy viděla!
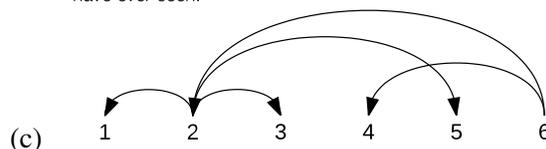   *The innuendos were really great... One of the best parodies I have ever seen!*

(c)

Figure 6: Induced dependency trees for three documents taken from Yelp (a,b) and the Czech Movies dataset (c). English translations are in italics.

into discourse units.

Figure 6 shows examples of document-level trees taken from Yelp and the Czech Movie dataset. In the first tree, most edges are examples of the "elaboration" discourse relation, i.e., the child presents

additional information about the parent. The second tree is non-projective, the edges connecting sentences 1 and 4 and 3 and 5 cross. The third review, perhaps due to its colloquial nature, is not entirely coherent. However, the model manages to link sentences 1 and 3 to sentence 2, i.e., the movie being discussed; it also relates sentence 6 to 4, both of which express highly positive sentiment.

5   Conclusions

In this paper we proposed a new model for representing documents while automatically learning rich structural dependencies. Our model normalizes intra-attention scores with the marginal probabilities of a non-projective dependency tree based on a matrix inversion process. Each operation in this process is differentiable and the model can be trained efficiently end-to-end, while inducing structural information. We applied this approach to model documents hierarchically, incorporating both sentence- and document-level structure. Experiments on sentence and document modeling tasks show that the representations learned by our model achieve competitive performance against strong comparison systems. Analysis of the induced tree structures revealed that they are meaningful, albeit different from linguistics ones, without ever exposing the model to linguistic annotations or an external parser.

Directions for future work are many and varied. Given appropriate training objectives (Linzen et al., 2016), it should be possible to induce linguistically meaningful dependency trees using the proposed attention mechanism. We also plan to explore how document-level trees can be usefully employed in summarization, e.g., as a means to represent or even extract important content.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of the ICLR Conference*.

James K. Baker. 1979. Trainable grammars for speech recognition. *The Journal of the Acoustical Society of America* 65(S1):S132–S132.

Regina Barzilay and Mirella Lapata. 2008. Modeling local coherence: An entity-based approach. *Computational Linguistics* 34(1):1–34.

Parminder Bhatia, Yangfeng Ji, and Jacob Eisenstein. 2015. Better document-level sentiment analysis from RST discourse parsing. In *Proceedings of the EMNLP Conference*. pages 2212–2218.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the EMNLP Conference*. pages 632–642.

Samuel R. Bowman, Jon Gauthier, Abhinav Rastogi, Raghav Gupta, Christopher D. Manning, and Christopher Potts. 2016. A fast unified model for parsing and sentence understanding. In *Proceedings of the ACL Conference*. pages 1466–1477.

Tomáš Brychcın and Ivan Habernal. 2013. Unsupervised improving of sentiment analysis using global target context. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing*. pages 122–128.

Lynn Carlson, Daniel Marcu, and Mary Ellen Okurowski. 2001. Building a discourse-tagged corpus in the framework of rhetorical structure theory. In *Proceedings of the Second SIGdial Workshop on Discourse and Dialogue*.

Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Si Wei, Hui Jiang, and Diana Inkpen. 2017. Enhanced LSTM for natural language inference. In *Proceedings of the ACL Conference*. pages 1657–1668.

Qian Chen, Xiaodan Zhu, Zhenhua Ling, Si Wei, and Hui Jiang. 2016. Distraction-based neural networks for modeling documents. In *Proceedings of the IJCAI Conference*. pages 2754–2760.

Jianpeng Cheng, Li Dong, and Mirella Lapata. 2016. Long short-term memory-networks for machine reading. In *Proceedings of the EMNLP Conference*. pages 551–561.

Yoeng-Jin Chu and Tseng-Hong Liu. 1965. On shortest arborescence of a directed graph. *Scientia Sinica* 14(10):1396.

Michał Daniluk, Tim Rocktäschel, Johannes Welbl, and Sebastian Riedel. 2017. Frustratingly short attention spans in neural language modeling. *Proceedings of the ICLR Conference* .

Qiming Diao, Minghui Qiu, Chao-Yuan Wu, Alexander J Smola, Jing Jiang, and Chong Wang. 2014. Jointly modeling aspects, ratings and sentiments for movie recommendation (JMARS). In *Proceedings of the ACM SIGKDD Conference*. pages 193–202.

John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research* 12(Jul):2121–2159.

Jack Edmonds. 1967. Optimum branchings. *Journal of Research of the national Bureau of Standards B* 71(4):233–240.

Vanessa Wei Feng and Graeme Hirst. 2012. Text-level discourse parsing with rich linguistic features. In *Proceedings of the ACL Conference*. pages 60–68.

Alex Graves, Abdel-Rahman Mohamed, and Geoffrey Hinton. 2013. Speech recognition with deep recurrent neural networks. In *Proceedings of the IEEE ICASSP Conference*. pages 6645–6649.

Katsuhiko Hayashi, Tsutomu Hirao, and Masaaki Nagata. 2016. Empirical comparison of dependency conversions for RST discourse trees. In *Pro-ceedings of the Annual Meeting of SIGDIAL*. page 128.

Tsutomu Hirao, Yasuhisa Yoshida, Masaaki Nishino, Norihito Yasuda, and Masaaki Nagata. 2013. Single-document summarization as a tree knapsack problem. In *Proceedings of the EMNLP Conference*. pages 1515–1520.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation* 9(8):1735–1780.

Yangfeng Ji and Noah Smith. 2017. Neural discourse structure for text categorization. In *Proceedings of the ACL Conference*.

Yoon Kim, Carl Denton, Luong Hoang, and Alexander M. Rush. 2017. Structured attention networks. In *Proceedings of the ICLR Conference*.

Terry Koo, Amir Globerson, Xavier Carreras Pérez, and Michael Collins. 2007. Structured prediction models via the matrix-tree theorem. In *Proceedings of the EMNLP Conference*. pages 141–150.

Alan Lee, Rashmi Prasad, Aravind Joshi, Nikhil Dinesh, and Bonnie Webber. 2006. Complexity of dependencies in discourse: Are dependencies in discourse more complex than in syntax. In *Proceedings of the International Workshop on Treebanks and Linguistic Theories*. page 12.

Sujian Li, Liang Wang, Ziqiang Cao, and Wenjie Li. 2014. Text-level discourse dependency parsing. In *Proceedings of the ACL Conference*. pages 25–35.

Ziheng Lin, Hwee Tou Ng, and Min-Yen Kan. 2011. Automatically evaluating text coherence using discourse relations. In *Proceedings of the ACL Conference*. pages 997–1006.

Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. 2016. Assessing the ability of LSTMs to learn syntax-sensitive dependencies. *Transactions of the Association for Computational Linguistics* 4:521–535.

Yang Liu and Mirella Lapata. 2017. Learning contextually informed representations for linear-time discourse parsing. In *Proceedings of the EMNLP Conference*. pages 1300–1309.

William C. Mann and Sandra A. Thompson. 1988. Rhetorical structure theory: Toward a functional theory of text organization. *Text-Interdisciplinary Journal for the Study of Discourse* 8(3):243–281.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP Natural Language Processing Toolkit. In *Proceedings of the ACL Conference (System Demonstrations)*. pages 55–60.

Ryan McDonald and Giorgio Satta. 2007. On the complexity of non-projective data-driven depen-

dency parsing. In *Proceedings of the 10th International Conference on Parsing Technologies*. pages 121–132.

Igor A. Melčuk. 1988. *Dependency Syntax: Theory and Practice*. State University of New York Press.

Mohsen Mesgar and Michael Strube. 2015. Graph-based coherence modeling for assessing readability. In *Proceedings of the 4th Joint Conference on Lexical and Computational Semantics*. pages 309–318.

Thomas Meyer and Bonnie Webber. 2013. Implicitation of discourse connectives in (machine) translation. In *Proceedings of the Workshop on Discourse in Machine Translation*. pages 19–26.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Proceedings of the NIPS Conference*. pages 3111–3119.

Alexander Miller, Adam Fisch, Jesse Dodge, Amir-Hossein Karimi, Antoine Bordes, and Jason Weston. 2016. Key-value memory networks for directly reading documents. In *Proceedings of the EMNLP Conference*. pages 1400–1409.

Ankur Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. 2016. A decomposable attention model for natural language inference. In *Proceedings of the EMNLP Conference*. pages 2249–2255.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the EMNLP Conference*. pages 1532–1543.

Rashmi Prasad, Nikhil Dinesh, Alan Lee, Eleni Miltsakaki, Livio Robaldo, Aravind K. Joshi, and Bonnie L. Webber. 2008. The Penn discourse TreeBank 2.0. In *LREC*.

Tim Rocktäschel, Edward Grefenstette, Karl Moritz Hermann, Tomáš Kočiský, and Phil Blunsom. 2016. Reasoning about entailment with neural attention. In *Proceedings of the ICLR Conference*.

Duyu Tang, Bing Qin, and Ting Liu. 2015a. Document modeling with gated recurrent neural network for sentiment classification. In *Proceedings of the EMNLP Conference*. pages 1422–1432.

Duyu Tang, Bing Qin, and Ting Liu. 2015b. Learning semantic representations of users and products for document level sentiment classification. In *Proceedings of the ACL Conference*. pages 1014–1023.

Louis Tesniére. 1959. *Éléments de Syntaxe Structurale*. Editions Klincksieck.

Matt Thomas, Bo Pang, and Lillian Lee. 2006. Get out the vote: Determining support or opposition from congressional floor-debate transcripts. In *Proceedings of the EMNLP Conference*. pages 327–335.

William Thomas Tutte. 1984. Graph theory.

Suzan Verberne, Lou Boves, Nelleke Oostdijk, and Peter-Arno Coppen. 2007. Discourse-based answering of why-questions. *Traitement Automatique des Language, Discours et Document: Traitements Automatics* 47(2):21–41.

Shuohang Wang and Jing Jiang. 2016. Learning natural language inference with LSTM. In *Proceedings of NAACL Conference*. pages 1442–1451.

Florian Wolf and Edward Gibson. 2006. *Coherence in Natural Language: Data Structures and Applications*. The MIT Press.

Pengtao Xie and Eric P. Xing. 2013. Integrating document clustering and topic modeling. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*. pages 694–703.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *International Conference on Machine Learning*. pages 2048–2057.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the NAACL Conference*. pages 1480–1489.

Dani Yogatama and Noah A. Smith. 2014. Linguistic structured sparsity in text categorization. In *Proceedings of the ACL Conference*. pages 786–796.

75