Structured Alignment Networks for Matching Sentences

Yang Liu*♠, Matt Gardner◇, and Mirella Lapata♠

♠University of Edinburgh
◇Allen Institute for Artificial Intelligence
yang.liu2@ed.ac.uk mattg@allenai.org mlap@inf.ed.ac.uk

Abstract

Many tasks in natural language processing involve comparing two sentences to compute some notion of relevance, entailment, or similarity. Typically, this comparison is done either at the word level or at the sentence level, with no attempt to leverage the inherent structure of the sentence. When sentence structure is used for comparison, it is obtained during a non-differentiable pre-processing step, leading to propagation of errors. We introduce a model of *structured alignments* between sentences, showing how to compare two sentences by matching their latent structures. Using a structured attention mechanism, our model matches candidate spans in the first sentence to candidate spans in the second sentence, simultaneously discovering the tree structure of each sentence. Our model is fully differentiable and trained only on the matching objective. We evaluate this model on two tasks, entailment detection and answer sentence selection, and find that modeling latent tree structures results in superior performance. Analysis of the learned sentence structures shows they can reflect some syntactic phenomena.

1 Introduction

There are many tasks in natural language processing that require matching two sentences: natural language inference (Bowman et al., 2015; Nangia et al., 2017) and paraphrase detection (Wang et al., 2017b) are classification tasks over sentence pairs, and question answering often requires an alignment between a question and a passage of text that may contain the answer (Tan et al., 2016a; Rajpurkar et al., 2016; Joshi et al., 2017).

Most neural models for these tasks perform comparisons between the two sentences either at the word level (Parikh et al., 2016), or at the sentence level (Bowman et al., 2015). Word-level comparisons ignore the inherent structure of the sentences being compared, at best relying on a recurrent neural network such as an LSTM (Hochreiter and Schmidhuber, 1997) to incorporate some amount of context from neighboring words into each word's representation. Sentence-level comparisons can incorporate the structure of each sentence individually (Bowman et al., 2016; Tai et al., 2015), but cannot easily compare substructures between the sentences, as these are all squashed into a single vector. Some models do incorporate sentence structure by comparing subtrees between the two sentences (Zhao et al., 2016; Chen et al., 2017), but require pipelined approaches where a parser is run in a non-differentiable preprocessing step, losing the benefits of end-to-end training.

In this paper we propose a method, which we call *structured alignment networks*, to perform comparisons between substructures in two sentences, in a more interpretable way, and without relying on an external, non-differentiable parser. We use a structured attention mechanism (Kim et al., 2017; Liu and Lapata, 2018) to compute a *structured alignment* between the two sentences, jointly learning a latent tree structure for each sentence and aligning spans between the two sentences.

Our method constructs a CKY chart for each sentence using the inside-outside algorithm (Manning et al., 1999), which is fully differentiable (Li and Eisner, 2009; Gormley et al., 2015). This chart has a node for each possible span in the sentence, and a score for the likelihood of that span being a constituent in a parse of the sentence, marginalized over all possible parses. We take these two charts and find alignments between them, representing each span in each sentence with structured attention over spans in the other sentence. These

[*] Work done during an internship at Allen Institute for Artificial Intelligence.

Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 1554–1564
Brussels, Belgium, October 31 - November 4, 2018.

span representations, weighted by the span's likelihood, are then used to compare the two sentences. In this way, we can perform comparisons between sentences by leveraging their internal structure in an end-to-end, fully differentiable model, trained only on one final objective. Our model helps obtain more precise representations of the sentence pair, with the learned tree structures and the alignment between them, and provides better interpretability, which most neural models lack in sentence matching tasks.

We evaluate this model on two sentence comparison datasets: SNLI (Bowman et al., 2015) and TREC-QA (Voorhees and Tice, 2000). We find that comparing sentences at the span level consistently outperforms comparing at the word level. Additionally, the learned sentence structures represent well-formed trees that reflect some syntactic phenomena.

2 Word-level Comparison Baseline

We first describe a common word-level comparison model, called *decomposable attention* (Parikh et al., 2016). This model was first proposed for the natural language inference task, but similar mechanisms have been used in many other tasks, such as for aligning question and passage words in the bi-directional attention model for question answering (Seo et al., 2017). This model serves as our main point of comparison, as our latent tree matching model simply replaces the word-level comparisons in decomposable attention model with span comparisons.

The decomposable attention model consists of three steps: *attend*, *compare*, and *aggregate*. As input, the model takes two sentences a and b represented by sequences of word embeddings $[\boldsymbol{a}_1, \cdots, \boldsymbol{a}_m]$ and $[\boldsymbol{b}_1, \cdots, \boldsymbol{b}_n]$. In the *attend* step, the model computes attention scores for each pair of words across the two input sentences and normalizes them as a soft alignment from a to b (and vice versa):

$$\boldsymbol{e}_{ij} = F_1(\boldsymbol{a}_i)^T F_1(\boldsymbol{b}_j) \tag{1}$$

$$\boldsymbol{B}_i = \sum_{j=1}^{n} \frac{exp(\boldsymbol{e}_{ij})}{\sum_{k=1}^{n} exp(\boldsymbol{e}_{ik})} \boldsymbol{b}_j \tag{2}$$

$$\boldsymbol{A}_j = \sum_{i=1}^{m} \frac{exp(\boldsymbol{e}_{ij})}{\sum_{k=1}^{m} exp(\boldsymbol{e}_{kj})} \boldsymbol{a}_i \tag{3}$$

where F_1 is a feed-forward neural network, $\boldsymbol{B}_i$ is the weighted summation of the words in b that are softly aligned to word $\boldsymbol{a}_i$ and vice versa for $\boldsymbol{A}_j$.

In the *compare* step, the input vectors $\boldsymbol{a}_i$ and $\boldsymbol{b}_j$ are concatenated with their corresponding attended vector $\boldsymbol{B}_i$ and $\boldsymbol{A}_j$, and fed into a feed-forward neural network, giving a comparison between each word and the words it aligns to in the other sentence:

$$\boldsymbol{v}_{ai} = F_2([\boldsymbol{a}_i, \boldsymbol{B}_i]) \quad \forall i \in [1, \cdots, m] \tag{4}$$

$$\boldsymbol{v}_{bj} = F_2([\boldsymbol{b}_j, \boldsymbol{A}_j]) \quad \forall j \in [1, \cdots, n] \tag{5}$$

The *aggregate* step is a simple summation of $\boldsymbol{v}_{ai}$ and $\boldsymbol{v}_{bj}$ for each word in sentence a and b, and the two resulting fixed-length vectors are concatenated and fed into a linear layer with $\boldsymbol{W}_y$ as the weight matrix, followed by a softmax layer for predicting the distribution $\boldsymbol{y}$:

$$\boldsymbol{v}_a = \sum_{i=1}^{m} \boldsymbol{v}_{ai} \quad \boldsymbol{v}_b = \sum_{j=1}^{n} \boldsymbol{v}_{bj} \tag{6}$$

$$\boldsymbol{y} = softmax(\boldsymbol{W}_y[\boldsymbol{v}_a, \boldsymbol{v}_b])) \tag{7}$$

The decomposable attention model completely ignores the order and context of words in the sequence. There are some efforts to strengthen the decomposable attention model with a recurrent neural network (Liu and Lapata, 2018) or intra-sentence attention (Parikh et al., 2016). However, these models amount to simply changing the input vectors $\boldsymbol{a}$ and $\boldsymbol{b}$, and still only perform a word-level alignment between the two sentences.

3 Structured Alignment Networks

Language is inherently tree structured, and the meaning of sentences comes largely from composing the meanings of subtrees (Chomsky, 2002). It is natural, then, to compare the meaning of two sentences by comparing their substructures (MacCartney and Manning, 2009). For example, when determining the relationship between two sentences in Figure 1, the ideal units of comparison are spans determined by subtrees: "*is in Seattle*" compared to "*based in Washington state*".

The challenge with comparing spans drawn from subtrees is that the tree structure of the sentence is latent and must be inferred, either during pre-processing or in the model itself. In this section we present a model that operates on the *latent* tree structure of each sentence, comparing *all possible* spans in one sentence with *all possible* spans in the second sentence, weighted by *how*

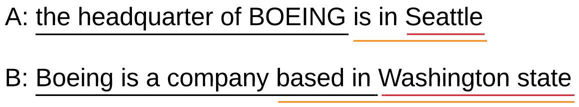

A: the headquarter of BOEING is in Seattle
B: Boeing is a company based in Washington state

Figure 1: Example span alignments of a sentence pair, where different colors indicate matching spans. Note that some spans overlap, which cannot happen in a single tree; our model considers *all possible* span comparisons, weighted by the spans' marginal likelihood.

likely each span is to appear as a constituent in a parse of the sentence. We use the non-terminal nodes of a binary constituency parse to represent spans. Because of this choice of representation, we can use the nodes in a CKY parsing chart to efficiently marginalize span likelihood over all possible parses for each sentence, and compare nodes in each sentence's chart.

3.1 Learning Latent Constituency Trees

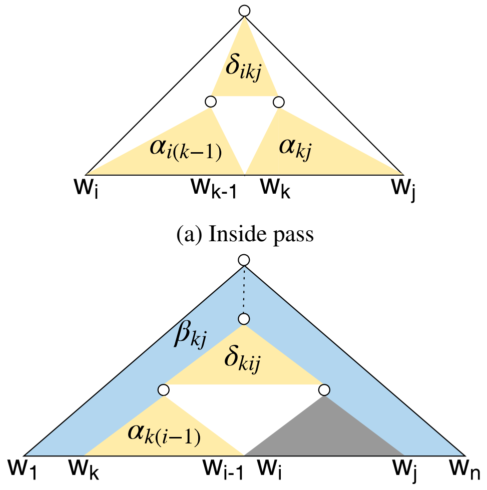
A constituency parser can be partially formalized as a graphical model with the following cliques (Klein and Manning, 2004): the latent variables $c_{ikj} \in 0, 1$ for all $i < j$, indicating whether the span from the i-th token to the j-th token ($span_{ij}$) is a constituency node built from the merging of sub-node $span_{ik}$ and $span_{(k+1)j}$. Given a sentence $x = [x_i, \cdots, x_n]$, the probability of a tree z is,

$$p(z|x) = \frac{\prod_{c_{ikj} \in z} p(c_{ikj} = 1)}{\sum_{z' \in Z} \prod_{c_{ikj} \in z'} p(c_{ikj} = 1)} \tag{8}$$

where Z represents all possible constituency trees for x.

The parameters for the graph-based CRF constituency parser are $\boldsymbol{\delta}_{ikj}$ reflecting the scores of $span_{ij}$ forming a binary constituency node with k as the splitting point. It is possible to calculate the marginal probability of each constituency node $p(c_{ijk} = 1|x)$ using the inside-outside algorithm (Klein and Manning, 2003). Although the inside-outside algorithm is constrained to generate a binary tree, this is not a severe limitation, as most structures can be easily binarized (Finkel et al., 2008).

In a typical constituency parser, the score $\boldsymbol{\delta}_{ikj}$ is parameterized according to the production rules of a grammar, e.g., with normalized categorical distributions for each non-terminal. Our unlabeled grammar effectively has only a single production rule, however, we parameterize these scores as bilinear functions operating on the representations of the two subtrees being merged. For the inside pass, as illustrated in Figure 2a, the inside score $\boldsymbol{\alpha}_{ij}$ for span from position i to j is marginalized over the splitting points k:



(a) Inside pass

(b) Outside pass

Figure 2: The inside-outside algorithm. (a) is the process for calculating the inside score $\boldsymbol{\alpha}_{ij}$. Three yellow spaces indicate $\boldsymbol{\alpha}_{i(k-1)}$, $\boldsymbol{\alpha}_{kj}$ and $\boldsymbol{\delta}_{ikj}$. (b) is a part of the process for calculating the outside score $\boldsymbol{\beta}_{ij}$, with target span $span_{ij}$ as the right child of a non-terminal. The blue space indicates $\boldsymbol{\beta}_{kj}$ and two yellow spaces indicate $\boldsymbol{\alpha}_{k(i-1)}$ and $\boldsymbol{\delta}_{kij}$.

$$\boldsymbol{\delta}_{ikj} = \boldsymbol{sp}_{ik}^{T} \boldsymbol{W} \boldsymbol{sp}_{(k+1)j} \tag{9}$$

$$\boldsymbol{\alpha}_{ij} = \sum_{i < k \leq j} \boldsymbol{\delta}_{ikj} \boldsymbol{\alpha}_{i(k-1)} \boldsymbol{\alpha}_{kj} \tag{10}$$

where $\boldsymbol{sp}_{ij} \in \mathbb{R}^{d}$ is the representation for the span, and $\boldsymbol{W} \in \mathbb{R}^{d*d}$ is the weight matrix. This process is calculated recursively from bottom to root, generating the score for each possible constituent.

For the outside pass, the outside score $\boldsymbol{\beta}_{ij}$ is:

$$\begin{aligned} \boldsymbol{\beta}_{ij} = &\sum_{1 \leq k < i} \boldsymbol{\delta}_{kij} \boldsymbol{\alpha}_{k(i-1)} \boldsymbol{\beta}_{kj} \\ &+ \sum_{j < k \leq n} \boldsymbol{\delta}_{ijk} \boldsymbol{\alpha}_{(j+1)k} \boldsymbol{\beta}_{ik} \end{aligned} \tag{11}$$

where the first term is the score for $span_{ij}$ being the right child on a non-terminal node and the second term is the score for $span_{ij}$ being the left child. In Figure 2b, we illustrate the outside process with the target span $span_{ij}$ being the right child of a non-terminal node. This process is calculated recursively from root to bottom.

The normalized marginal probability $\boldsymbol{\rho}_{ij}$ for each span $span_{ij}$, where $1 \leq i < n, i < j \leq n$ can be calculated by:

$$\boldsymbol{\rho}_{ij} = \boldsymbol{\alpha}_{ij}\boldsymbol{\beta}_{ij}/\boldsymbol{\alpha}_{0n} \tag{12}$$

To compute the representations of all possible spans, we use Long Short-Term Memory Neural Networks (LSTMs; Hochreiter and Schmidhuber 1997) with max-pooling and minus features (Cross and Huang, 2016; Liu and Lapata, 2017). We represent each sentence as a sequence of word embeddings $[\boldsymbol{w}_{sos}, \boldsymbol{w}_1, \cdots, \boldsymbol{w}_t, \cdots, \boldsymbol{w}_n, \boldsymbol{w}_{eos}]$ and run a bidirectional LSTM to obtain the output vectors. $\boldsymbol{h}_t = [\overrightarrow{\boldsymbol{h}}_t, \overleftarrow{\boldsymbol{h}}_t]$ is the output vector for the t^{th} word, and $\overrightarrow{\boldsymbol{h}}_t$ and $\overleftarrow{\boldsymbol{h}}_t$ are the output vectors from the forward and backward directions, respectively. We represent a constituent from position i to j with a span vector $\boldsymbol{sp}_{ij}$:

$$\boldsymbol{sp}_{ij}^{maxpool} = \max(\boldsymbol{h}_i, \cdots, \boldsymbol{h}_j) \tag{13}$$

$$\boldsymbol{sp}_{ij}^{minus} = [\overrightarrow{\boldsymbol{h}}_j - \overrightarrow{\boldsymbol{h}}_{i-1}, \overleftarrow{\boldsymbol{h}}_i - \overleftarrow{\boldsymbol{h}}_{j+1}] \tag{14}$$

$$\boldsymbol{sp}_{ij} = [\boldsymbol{sp}_{ij}^{maxpool}, \boldsymbol{sp}_{ij}^{minus}] \tag{15}$$

where $\max(\boldsymbol{x}_i, \cdots, \boldsymbol{x}_j)$ is the max-pooling operation over the sequence of output vectors within this constituent.

After applying the parsing process on two sentences, we obtain the marginal probabilities for all potential spans of the two constituency trees, which can then be used for aligning.

3.2 Learning Structured Alignments

After learning latent constituency trees for each sentence, we are able to perform span-level comparisons between the two sentences, instead of the word-level comparisons done by the decomposable attention model. The structure of these two comparison models is the same, but the basic elements of our structured alignment model are spans instead of words, and the marginal probabilities obtained from the inside-outside algorithm are used as a *re-normalization* value for incorporating structural information into the alignments.

For sentence a, we have the representation $\boldsymbol{sp}_{ij}^a$ for each $span_{ij}$ and its marginal probability $\boldsymbol{\rho}_{ij}^a$. And for sentence b, we also get $\boldsymbol{sp}_{ij}^b$ and $\boldsymbol{\rho}_{ij}^b$. The attention scores are computed between all pairs of spans across the two sentences, and the attended vectors can be calculated as:

$$\boldsymbol{e}_{ij,kl} = F_1(\boldsymbol{sp}_{ij}^a)^T F_1(\boldsymbol{sp}_{kl}^b) \tag{16}$$

$$\boldsymbol{B}_{ij} = \sum_{k=1}^{n}\sum_{l=k}^{n} \frac{exp(\boldsymbol{e}_{ij,kl} + ln(\boldsymbol{\rho}_{kl}^b))}{\sum_{s=1}^{n}\sum_{t=s}^{n} exp(\boldsymbol{e}_{ij,st} + ln(\boldsymbol{\rho}_{st}^b))} \boldsymbol{sp}_{kl}^b \tag{17}$$

$$\boldsymbol{A}_{kl} = \sum_{i=1}^{m}\sum_{j=i}^{m} \frac{exp(\boldsymbol{e}_{ij,kl} + ln(\boldsymbol{\rho}_{ij}^a))}{\sum_{s=1}^{m}\sum_{t=s}^{m} exp(\boldsymbol{e}_{st,kl} + ln(\boldsymbol{\rho}_{st}^a))} \boldsymbol{sp}_{ij}^a \tag{18}$$

Then, the span vectors are concatenated with the attended vectors and fed into a feed-forward neural network:

$$\boldsymbol{v}_{ij}^a = F_2([\boldsymbol{sp}_{ij}^a, \boldsymbol{B}_{ij}]) \tag{19}$$

$$\boldsymbol{v}_{kl}^b = F_2([\boldsymbol{sp}_{kl}^b, \boldsymbol{A}_{kl}]) \tag{20}$$

To aggregate these vectors, instead of using direct summation, we apply weighted summation with the marginal probabilities as weights:

$$\boldsymbol{v}_a = \sum_{i=1}^{m}\sum_{j=i}^{m} \boldsymbol{\rho}_{ij}^a \boldsymbol{v}_{ij}^a; \boldsymbol{v}_b = \sum_{k=1}^{n}\sum_{l=1}^{n} \boldsymbol{\rho}_{kl}^b \boldsymbol{v}_{kl}^b \tag{21}$$

where $\boldsymbol{\rho}^a$ and $\boldsymbol{\rho}^b$ work like the self-attention mechanism of (Lin et al., 2017) to replace the summation pooling step. We use a softmax function to compute the predicted distribution $\boldsymbol{y}$ of the input sentence pair:

$$\boldsymbol{y} = softmax(\boldsymbol{W}_y[\boldsymbol{v}_a, \boldsymbol{v}_b]) \tag{22}$$

4 Experiments

We evaluate our structured alignment model on two natural language matching tasks: question answering as sentence selection and natural language inference. We view our approach as a module for replacing the widely-used word-level alignment which can be plugged into other neural models. For that reason, our experiments are not intended to show performance improvements over state-of-the-art neural network architectures. Rather our evaluation studies aim to address three questions: (a) whether our methods can be trained effectively in an end-to-end fashion; (b) whether they yield improvements over standard word-level alignment models; and (c) whether they can learn plausible latent constituency tree structures.

Models	MAP	MRR
Word-level Attention	0.764	0.842
Simple Span Alignment	0.772	0.851
Simple Span Alignment + External Parser	0.780	0.846
Structured Alignment (Shared Parameters)	0.780	0.860
Structured Alignment (Separated Parameters)	**0.786**	**0.860**
QA-LSTM (Tan et al., 2016b)	0.730	0.824
Attentive Pooling Network (Santos et al., 2016)	0.753	0.851
Pairwise Word Interaction (He and Lin, 2016)	0.777	0.836
Lexical Decomposition and Composition (Wang et al., 2016)	0.771	0.845
Noise-Contrastive Estimation (Rao et al., 2016)	0.801	**0.877**
BiMPM (Wang et al., 2017b)	**0.802**	0.875

Table 1: Results of our models (top) and previously proposed systems (bottom) on the TREC-QA test set.

For both tasks, we initialize our model with 300D 840B GloVe word embeddings (Pennington et al., 2014). The hidden size for the BiLSTM is 150. The feed-forward networks F_1 and F_2 are two-layer perceptrons with ReLU as the hidden activation function and the size of the hidden and output layers is set to 300. All hyperparameters are selected based on the model's performance on the development set.

4.1 Answer Sentence Selection

We first study the effectiveness of our model for answer sentence selection tasks. Given a question, answer sentence selection aims to rank a list of candidate answer sentences based on their relatedness to the question. We experiment on the TREC-QA dataset (Wang et al., 2007), in which all questions with only positive or negative answers are removed. This leaves us with 1,162 training questions, 65 development questions and 68 test questions. Experimental results are listed in Table 1. We measure performance by the mean average precision (MAP) and mean reciprocal rank (MRR) using the standard TREC evaluation script.

In the first block of Table 1, we compare our model and variants thereof against several baselines. The first baseline is the *Word-level Decomposable Attention* model strengthened with a bidirectional LSTM for obtaining a contextualized representation for each word. The second baseline is a *Simple Span Alignment* model; we use an MLP layer over the LSTM outputs to calculate the unnormalized scores and replace the inside-outside algorithm with a simple softmax function to obtain the probability distribution over all candidate spans. We also introduce a pipelined baseline where we extract constituents from trees parsed by the CoreNLP (Manning et al., 2014) constituency parser, and use the *Simple Span Alignment* model to only align these constituents.

As shown in Table 1, we use two variants of the *Structured Alignment* model, since the structure of the question and the answer sentence may be different; the first model shares parameters across the question and the answer for computing the structures, while the second one uses separate parameters. We view the sentence selection task as a binary classification problem and the final ranking is based on the predicted probability of the sentence containing the correct answer (positive label). We apply dropout to the output of the BiLSTM with dropout ratio set to 0.2. All parameters (including word embeddings) are updated with AdaGrad (Duchi et al., 2011), and the learning rate is set to 0.05.

Table 1 (second block) also reports the performance of various comparison systems and state-of-the-art models. As can be seen, on both MAP and MRR metrics, structured alignment models perform better than the decomposable attention model, showing that structural bias is helpful for matching a question to the correct answer sentence. We also observe that using separate parameters achieves higher scores on both metrics. The simple span alignment model obtains results similar to the decomposable attention model, suggesting that the shallow softmax distribution is ineffective for capturing structural information and may even introduce redundant noise. The pipelined model with an external parser also slightly im-

Models	Accuracy	# Parameters
Word-level Attention	85.8	1.1M
Simple Span Alignment	85.4	1.26M
Simple Span Alignment + External Parser	85.6	1.17M
Structured Alignment	**86.3**	1.44M
Classifier with handcrafted features (Bowman et al., 2015)	78.2	-
LSTM encoders (Bowman et al., 2015)	80.6	3.0M
LSTM with inter-attention (Rocktäschel et al., 2016)	83.5	252K
Matching LSTMs (Wang and Jiang, 2015)	86.1	1.9M
LSTMN with deep attention fusion (Cheng et al., 2016)	86.3	3.4M
Enhanced BiLSTM Inference Model (Chen et al., 2016)	**88.0**	4.3M
Densely Interactive Inference Network (Gong et al., 2017)	**88.0**	-

Table 2: Test accuracy (%) on the SNLI dataset. Wherever available we also provide the number of parameters (excluding embeddings).

proves upon the baseline, but still cannot outperform the end-to-end trained structured alignment model which achieves results comparable with several strong baselines with fewer parameters. As mentioned earlier, our model could be used as a plug-in component for other more complex models, and may boost their performance by modeling the latent structures. At the same time, the structured alignment can provide better interpretability for sentence matching tasks, which is a defect of most neural models.

4.2 Natural Language Inference

The second task we consider is natural language inference, where the input is a pair of premise and hypothesis sentences, and the goal is to predict whether the premise entails the hypothesis, contradicts the hypothesis, or neither. For this task, we use the Stanford NLI dataset (Bowman et al., 2015). After removing sentences with unknown labels, we obtain 549,367 pairs for training, 9,842 for development and 9,824 for testing.

We compare our model against the same baselines used in the question answering task. All parameters (including word embeddings) are updated with AdaGrad (Duchi et al., 2011), and the learning rate is set to 0.05. Dropout is used with ratio 0.2. The structured alignment model in this experiment uses shared parameters for computing latent tree structures, since both the premise and hypothesis are declarative sentences.

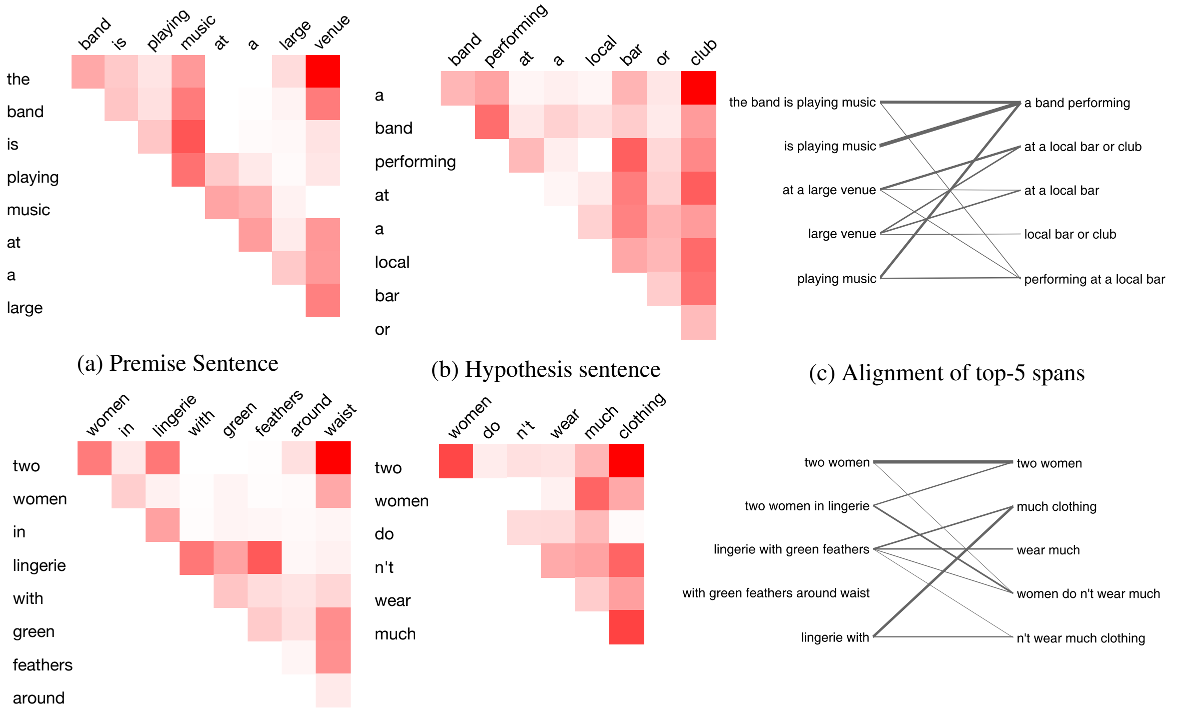
The results of our experiments are shown in Table 2. Similar to the answer selection task, the tree matching model outperforms the decomposable model. Our structured alignment model gains 0.5% in accuracy over the baseline word-level comparison model without any additional annotation, simply from introducing a structural bias in the alignment between the sentences. Simple span alignment, however, is not helpfult and even slightly degrades the performance over the word-level model.

4.3 Analysis of Learned Tree Structures

In this section, we give a brief qualitative analysis of the learned tree structures. We present the CKY charts for two randomly-selected sentence pairs in the SNLI test set in Figure 3. Recall that the CKY chart shows the likelihood of each span appearing as a constituent in the parse of the sentence, marginalized over all possible parses. By visualizing these span probabilities, we can see that the model learns structures which correspond to known syntactic structures.

In subfigure (a), we can see that *band is playing* is a very-likely span, as is *at a large venue*. In subfigure (b), the phrases *performing at a local bar* and *at a local bar or club* also receive high probabilities. For the second sentence pair, we see that the model can even resolve some attachment ambiguities correctly. The prepositional phrase *with green feathers*, has a very low score for being attached to *women*. Instead, the model prefers to attach it to *lingerie*, forming the span *lingerie with green feathers*. We also present the top-5 spans and their alignments in subfigures (c) and (d), which can be used to interpret model decisions for sentence matching tasks.

The analysis above and our experimental results in the previous section suggest that our



(a) Premise Sentence (b) Hypothesis sentence (c) Alignment of top-5 spans

(d) Premise Sentence (e) Hypothesis sentence (f) Alignment of top-5 spans

Figure 3: (a), (b), (d), and (e) are CKY charts showing marginalized span probabilities for sentence pairs in the SNLI test set. Each cell uses depth of the color to represent the probability of the span (from the i-th word to the j-th word) forming a proper constituent. (c) and (f) are the alignments of the top-5 spans from hypothesis sentence to the premise sentence, where the boldness of the lines indicates the probability of spans being aligned.

model is able to learn tree structures which are closely related to syntax, and in addition reflect the semantic-level characteristics of the task at hand. In both question answering and natural language inference tasks, we observe that structured alignment leads to performance improvements over word-level models. This is in contrast to prior work (Williams et al., 2017), where the discovery of tree structures based on a semantic objective is not helpful. Although we use the same supervision signal in our model, a difference between the two approaches is that they are trying to learn tree structures for each sentence *independently*, performing comparisons at the sentence level only. Comparing spans directly forces the model to induce trees with comparable constituents, giving the model a stronger inductive bias.

Although our main goal is not to induce a grammar, we perform some simple experiments to compare the learned latent trees with parser-generated ones. We parse the sentences in both test-sets with the CoreNLP (Manning et al., 2014) constituency parser to obtain silver trees. Based on the parsing part of the trained structured alignment model, we compute the marginal probabilities of test sentences and feed them into CKY algorithm (Younger, 1967) to find the most likely constituency trees. We then convert both silver and latent trees to sets of constituent brackets, and calculate the accuracy of the learned brackets against the silver parses. We use different combinations of training- and test-sets to examine the transferability of the learned tree structures. The results are shown in Table 3. We can see that although our model does not have any tree-structured input during training, it can still outperform the left-branching (LB) and right-branching baselines (RB) and achieve some consistency with the parser generated trees.

5 Related Work

Sentence comparison models The Stanford natural language inference dataset (Bowman et al., 2015), and the expanded multi-genre natural language inference dataset (Nangia et al., 2017), are

trained on tested on	SNLI	TREC	LB	RB
SNLI	**15.1**	10.7	12.8	6.0
TREC	12.3	**11.4**	10.5	3.2

Table 3: Brackets accuracy of latent learned trees against silver trees from CoreNLP parser. We show the transferability of the learned parser by applying it on a test-set different from the training-set. For example, we train the structured alignment model on the TREC-QA data, and apply it on SNLI to obtain the tree distributions. LB and RB are left- and right-branching baselines.

the most well-known recent sentence comparison tasks. The literature on this comparison task is far too extensive to include here, although the recent shared task on Multi-NLI gives a good survey of sentence-level comparison models (Nangia et al., 2017). Some of these models use sentence structures, which are obtained either in a latent fashion (Bowman et al., 2016) or during preprocessing (Zhao et al., 2016), but they squash all of the structure into a single vector, losing the ability to easily compare substructures between the two sentences.

For models doing a word-level comparison, the decomposable attention model (Parikh et al., 2016), which we have discussed already in this paper, is the most salient example, although many similar models exist in the literature (Chen et al., 2017; Wang et al., 2017b). The idea of word-level alignments between a question and a passage is also pervasive in the recent question answering literature (Seo et al., 2017; Wang et al., 2017a).

Finally, and most similar to our approach, several models have been proposed that directly compare subtrees between two sentences (Chen et al., 2017; Zhao et al., 2016). However, all of these models are pipelined; they obtain the sentence structure in a non-differentiable preprocessing step, losing the benefits of end-to-end training. Ours is the first model to allow comparison between *latent* tree structures, trained end-to-end on the comparison objective.

Structured attention While it has long been known that inference in graphical models is differentiable (Li and Eisner, 2009; Domke, 2011), and using inference in, e.g., a CRF (Lafferty et al., 2001) as the last layer in a neural network is common practice (Liu and Lapata, 2017; Lample et al., 2016), the use of inference algorithms as intermediate layers in end-to-end neural networks is a recent development. Kim et al. (2017) were the first to use inference to compute structured attentions over latent sentence variables, inducing tree structures trained on the end-to-end objective. Liu and Lapata (2018) showed how to do this more efficiently, although their work is still limited to structured attention over a single sentence. Our model is the first to include latent structured alignments between two sentences.

Grammar Induction Unsupervised grammar induction is a well-studied problem (Cohen and Smith, 2009). The most recent work in this direction was the Neural E-DMV model of Jiang et al. (2016). While our goal is not to induce a grammar, we do produce a probabilistic grammar as a byproduct of our model. Our results suggest that training on more complex objectives may be a good way to pursue grammar induction in the future; forcing the model to construct consistent, comparable subtrees between the two sentences is a strong signal for grammar induction. Very recently, a few models attempt to infer latent dependency tree structures with neural models in sentence modeling tasks (Yogatama et al., 2017; Choi et al., 2018).

6 Conclusions

In this paper we have considered the problem of comparing two sentences in natural language processing models. We have shown how to move beyond word- and sentence-level comparison to comparing spans between two sentences, without the need for an external parser. Through experiments on sentence comparison datasets, we have seen that span comparisons consistently outperform word-level comparisons, with no additional supervision. The proposed model can be trained effectively, in an end-to-end fashion and is able to induce plausible tree structures.

Our results have several implications for future work. First, the success of span comparisons over word-level comparisons suggests that it may be advantageous to include such comparisons in more complex models, either for comparing two sentences directly, or as intermediate parts of models for more complex tasks, such as reading comprehension. Second, our model's ability to infer trees from a semantic objective is intriguing, and suggestive of future opportunities in grammar induction research. The use of the inside-outside algorithm unavoidably renders the full model er (by 5–8 times) compared to the decomposable attention model. We hope to find a more efficient way to accelerate this dynamic programming method on a GPU.

Acknowledgments This research is supported by a Google PhD Fellowship to the first author. We acknowledge the financial support of the European Research Council (Lapata; award number 681760).

References

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 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal.

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 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1466–1477, Berlin, Germany.

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 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1657–1668, Vancouver, Canada.

Qian Chen, Xiaodan Zhu, Zhenhua Ling, Si Wei, and Hui Jiang. 2016. Enhancing and combining sequential and tree LSTM for natural language inference. *arXiv preprint arXiv:1609.06038*.

Jianpeng Cheng, Li Dong, and Mirella Lapata. 2016. Long short-term memory-networks for machine reading. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 551–561, Austin, Texas.

Jihun Choi, Kang Min Yoo, and Sang goo Lee. 2018. Learning to compose task-specific tree structures. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*, pages 5094–5101, New Orleans, Louisiana.

Noam Chomsky. 2002. *Syntactic structures*. Walter de Gruyter.

Shay Cohen and Noah A. Smith. 2009. Shared logistic normal distributions for soft parameter tying in unsupervised grammar induction. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 74–82, Boulder, Colorado.

James Cross and Liang Huang. 2016. Span-based constituency parsing with a structure-label system and provably optimal dynamic oracles. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1–11, Austin, Texas.

Justin Domke. 2011. Parameter learning with truncated message-passing. In *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2937–2943, Colorado Springs, Colorado.

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

Jenny Rose Finkel, Alex Kleeman, and Christopher D. Manning. 2008. Efficient, feature-based, conditional random field parsing. In *Proceedings of ACL-08: HLT*, pages 959–967, Columbus, Ohio.

Yichen Gong, Heng Luo, and Jian Zhang. 2017. Natural language inference over interaction space. *arXiv preprint arXiv:1709.04348*.

Matthew Gormley, Mark Dredze, and Jason Eisner. 2015. Approximation-aware dependency parsing by belief propagation. *Transactions of the Association for Computational Linguistics*, 3:489–501.

Hua He and Jimmy Lin. 2016. Pairwise word interaction modeling with deep neural networks for semantic similarity measurement. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 937–948, San Diego, California.

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

Yong Jiang, Wenjuan Han, and Kewei Tu. 2016. Unsupervised neural dependency parsing. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 763–771, Austin, Texas.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada.

Yoon Kim, Carl Denton, Luong Hoang, and Alexander M Rush. 2017. Structured attention networks. In *Proceedings of the 5th International Cofnerence on Learning Representations*, Toulon, France.

Dan Klein and Christopher Manning. 2004. Corpus-based induction of syntactic structure: Models of dependency and constituency. In *Proceedings of the 42nd Meeting of the Association for Computational Linguistics, Main Volume*, pages 478–485, Barcelona, Spain.

Dan Klein and Christopher D Manning. 2003. Fast exact inference with a factored model for natural language parsing. In *Advances in Neural Information Processing Systems 15*, pages 3–10. MIT Press.

John Lafferty, Andrew McCallum, and Fernando C.N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the 18th International Conference on Machine Learning*, pages 282–289, Williamstown, Massachusetts.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 260–270, San Diego, California.

Zhifei Li and Jason Eisner. 2009. First- and second-order expectation semirings with applications to minimum-risk training on translation forests. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 40–51, Singapore.

Zhouhan Lin, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. 2017. A structured self-attentive sentence embedding. *arXiv preprint arXiv:1703.03130*.

Yang Liu and Mirella Lapata. 2017. Learning contextually informed representations for linear-time discourse parsing. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1289–1298, Copenhagen, Denmark.

Yang Liu and Mirella Lapata. 2018. Learning structured text representations. *Transactions of the Association for Computational Linguistics*, 6:63–75.

Bill MacCartney and Christopher D Manning. 2009. An extended model of natural logic. In *Proceedings of the international conference on computational semantics*.

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60, Baltimore, Maryland.

Christopher D Manning, Hinrich Schütze, et al. 1999. *Foundations of statistical natural language processing*, volume 999. MIT Press.

Nikita Nangia, Adina Williams, Angeliki Lazaridou, and Samuel R Bowman. 2017. The repeval 2017 shared task: Multi-genre natural language inference with sentence representations. *arXiv preprint arXiv:1707.08172*.

Ankur Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. 2016. A decomposable attention model for natural language inference. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2249–2255, Austin, Texas.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas.

Jinfeng Rao, Hua He, and Jimmy Lin. 2016. Noise-contrastive estimation for answer selection with deep neural networks. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management Conference*, pages 1913–1916, Indianapolis, Indiana.

Tim Rocktäschel, Edward Grefenstette, Karl Moritz Hermann, Tomáš Kočiskỳ, and Phil Blunsom. 2016. Reasoning about entailment with neural attention. In *Proceedings of the 4th International Cofnerence on Learning Representations*, San Juan, Puerto Rico.

Cicero dos Santos, Ming Tan, Bing Xiang, and Bowen Zhou. 2016. Attentive pooling networks. *arXiv preprint arXiv:1602.03609*.

Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2017. Bidirectional attention flow for machine comprehension. In *Proceeding of the 5th International Cofnerence on Learning Representations*, Toulon, France.

Kai Sheng Tai, Richard Socher, and Christopher D. Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1556–1566, Beijing, China.

Ming Tan, Cicero dos Santos, Bing Xiang, and Bowen Zhou. 2016a. Improved representation learning for question answer matching. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 464–473, Berlin, Germany.

Ming Tan, Cicero dos Santos, Bing Xiang, and Bowen Zhou. 2016b. LSTM-based deep learning models for non-factoid answer selection. In *Proceedings of the ICLR 2016 Workshop Track*, San Juan, Puerto Rico.

Ellen M Voorhees and Dawn M Tice. 2000. Building a question answering test collection. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 200–207, Athens, Greece.

Mengqiu Wang, Noah A. Smith, and Teruko Mitamura. 2007. What is the Jeopardy model? a quasi-synchronous grammar for QA. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 22–32, Prague, Czech Republic.

Shuohang Wang and Jing Jiang. 2015. Learning natural language inference with LSTM. *arXiv preprint arXiv:1512.08849*.

Wenhui Wang, Nan Yang, Furu Wei, Baobao Chang, and Ming Zhou. 2017a. Gated self-matching networks for reading comprehension and question answering. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 189–198, Vancouver, Canada.

Zhiguo Wang, Wael Hamza, and Radu Florian. 2017b. Bilateral multi-perspective matching for natural language sentences. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pages 4144–4150, Melbourne, Australia.

Zhiguo Wang, Haitao Mi, and Abraham Ittycheriah. 2016. Sentence similarity learning by lexical decomposition and composition. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1340–1349, Osaka, Japan.

Adina Williams, Andrew Drozdov, and Samuel R Bowman. 2017. Learning to parse from a semantic objective: It works. is it syntax? *arXiv preprint arXiv:1709.01121*.

Dani Yogatama, Phil Blunsom, Chris Dyer, Edward Grefenstette, and Wang Ling. 2017. Learning to compose words into sentences with reinforcement learning. In *Proceedings of the 5th International Cofnerence on Learning Representations*, Toulon, France.

Daniel H Younger. 1967. Recognition and parsing of context-free languages in time n^3. *Information and Control*, 10:189–208.

Kai Zhao, Liang Huang, and Mingbo Ma. 2016. Textual entailment with structured attentions and composition. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2248–2258, Osaka, Japan.