REASONING ABOUT ENTAILMENT WITH NEURAL ATTENTION

Tim Rocktäschel
University College London
t.rocktaschel@cs.ucl.ac.uk

Edward Grefenstette & Karl Moritz Hermann
Google DeepMind
{etg,kmh}@google.com

Tomáš Kočiský & Phil Blunsom
Google DeepMind & University of Oxford
{tkocisky,pblunsom}@google.com

ABSTRACT

While most approaches to automatically recognizing entailment relations have used classifiers employing hand engineered features derived from complex natural language processing pipelines, in practice their performance has been only slightly better than bag-of-word pair classifiers using only lexical similarity. The only attempt so far to build an end-to-end differentiable neural network for entailment failed to outperform such a simple similarity classifier. In this paper, we propose a neural model that reads two sentences to determine entailment using long short-term memory units. We extend this model with a word-by-word neural attention mechanism that encourages reasoning over entailments of pairs of words and phrases. Furthermore, we present a qualitative analysis of attention weights produced by this model, demonstrating such reasoning capabilities. On a large entailment dataset this model outperforms the previous best neural model and a classifier with engineered features by a substantial margin. It is the first generic end-to-end differentiable system that achieves state-of-the-art accuracy on a textual entailment dataset.

1 INTRODUCTION

The ability to determine the semantic relationship between two sentences is an integral part of machines that understand and reason with natural language. Recognizing textual entailment (RTE) is the task of determining whether two natural language sentences are (i) contradicting each other, (ii) not related, or whether (iii) the first sentence (called *premise*) entails the second sentence (called *hypothesis*). This task is important since many natural language processing (NLP) problems, such as information extraction, relation extraction, text summarization or machine translation, rely on it explicitly or implicitly and could benefit from more accurate RTE systems (Dagan et al., 2006).

State-of-the-art systems for RTE so far relied heavily on engineered NLP pipelines, extensive manual creation of features, as well as various external resources and specialized subcomponents such as negation detection (e.g. Lai and Hockenmaier, 2014; Jimenez et al., 2014; Zhao et al., 2014; Beltagy et al., 2015). Despite the success of neural networks for paraphrase detection (e.g. Socher et al., 2011; Hu et al., 2014; Yin and Schütze, 2015), end-to-end differentiable neural architectures failed to get close to acceptable performance for RTE due to the lack of large high-quality datasets. An end-to-end differentiable solution to RTE is desirable, since it avoids specific assumptions about the underlying language. In particular, there is no need for language features like part-of-speech tags or dependency parses. Furthermore, a generic sequence-to-sequence solution allows to extend the concept of capturing entailment across any sequential data, not only natural language.

Recently, Bowman et al. (2015) published the Stanford Natural Language Inference (SNLI) corpus accompanied by a neural network with long short-term memory units (LSTM, Hochreiter and Schmidhuber, 1997), which achieves an accuracy of 77.6% for RTE on this dataset. It is the first time a generic neural model without hand-crafted features got close to the accuracy of a simple lexicalized classifier with engineered features for RTE. This can be explained by the high quality

and size of SNLI compared to the two orders of magnitude smaller and partly synthetic datasets so far used to evaluate RTE systems. Bowman et al.'s LSTM encodes the premise and hypothesis as dense fixed-length vectors whose concatenation is subsequently used in a multi-layer perceptron (MLP) for classification. In contrast, we are proposing an attentive neural network that is capable of reasoning over entailments of pairs of words and phrases by processing the hypothesis conditioned on the premise.

Our contributions are threefold: (i) We present a neural model based on LSTMs that reads two sentences in one go to determine entailment, as opposed to mapping each sentence independently into a semantic space (§2.2), (ii) We extend this model with a neural word-by-word attention mechanism to encourage reasoning over entailments of pairs of words and phrases (§2.4), and (iii) We provide a detailed qualitative analysis of neural attention for RTE (§4.1). Our benchmark LSTM achieves an accuracy of 80.9% on SNLI, outperforming a simple lexicalized classifier tailored to RTE by 2.7 percentage points. An extension with word-by-word neural attention surpasses this strong benchmark LSTM result by 2.6 percentage points, setting a new state-of-the-art accuracy of 83.5% for recognizing entailment on SNLI.

2 METHODS

In this section we discuss LSTMs (§2.1) and describe how they can be applied to RTE (§2.2). We introduce an extension of an LSTM for RTE with neural attention (§2.3) and word-by-word attention (§2.4). Finally, we show how such attentive models can easily be used for attending both ways: over the premise conditioned on the hypothesis and over the hypothesis conditioned on the premise (§2.5).

2.1 LSTMS

Recurrent neural networks (RNNs) with long short-term memory (LSTM) units (Hochreiter and Schmidhuber, 1997) have been successfully applied to a wide range of NLP tasks, such as machine translation (Sutskever et al., 2014), constituency parsing (Vinyals et al., 2014), language modeling (Zaremba et al., 2014) and recently RTE (Bowman et al., 2015). LSTMs encompass memory cells that can store information for a long period of time, as well as three types of gates that control the flow of information into and out of these cells: input gates (Eq. 2), forget gates (Eq. 3) and output gates (Eq. 4). Given an input vector $\mathbf{x}_t$ at time step t, the previous output $\mathbf{h}_{t-1}$ and cell state $\mathbf{c}_{t-1}$, an LSTM with hidden size k computes the next output $\mathbf{h}_t$ and cell state $\mathbf{c}_t$ as

$$\mathbf{H} = \begin{bmatrix} \mathbf{x}_t \\ \mathbf{h}_{t-1} \end{bmatrix} \quad (1)$$

$$\mathbf{i}_t = \sigma(\mathbf{W}^i\mathbf{H} + \mathbf{b}^i) \quad (2)$$

$$\mathbf{f}_t = \sigma(\mathbf{W}^f\mathbf{H} + \mathbf{b}^f) \quad (3)$$

$$\mathbf{o}_t = \sigma(\mathbf{W}^o\mathbf{H} + \mathbf{b}^o) \quad (4)$$

$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \tanh(\mathbf{W}^c\mathbf{H} + \mathbf{b}^c) \quad (5)$$

$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{c}_t) \quad (6)$$

where $\mathbf{W}^i, \mathbf{W}^f, \mathbf{W}^o, \mathbf{W}^c \in \mathbb{R}^{2k \times k}$ are trained matrices and $\mathbf{b}^i, \mathbf{b}^f, \mathbf{b}^o, \mathbf{b}^c \in \mathbb{R}^k$ trained biases that parameterize the gates and transformations of the input, σ denotes the element-wise application of the sigmoid function and $\odot$ the element-wise multiplication of two vectors.

2.2 RECOGNIZING TEXTUAL ENTAILMENT WITH LSTMS

LSTMs can readily be used for RTE by independently encoding the premise and hypothesis as dense vectors and taking their concatenation as input to an MLP classifier (Bowman et al., 2015). This demonstrates that LSTMs can learn semantically rich sentence representations that are suitable for determining textual entailment.

2.2.1 CONDITIONAL ENCODING

In contrast to learning sentence representations, we are interested in neural models that read both sentences to determine entailment, thereby reasoning over entailments of pairs of words and phrases. Figure 1 shows the high-level structure of this model. The premise (left) is read by an LSTM. A second LSTM with different parameters is reading a delimiter and the hypothesis (right), but its memory

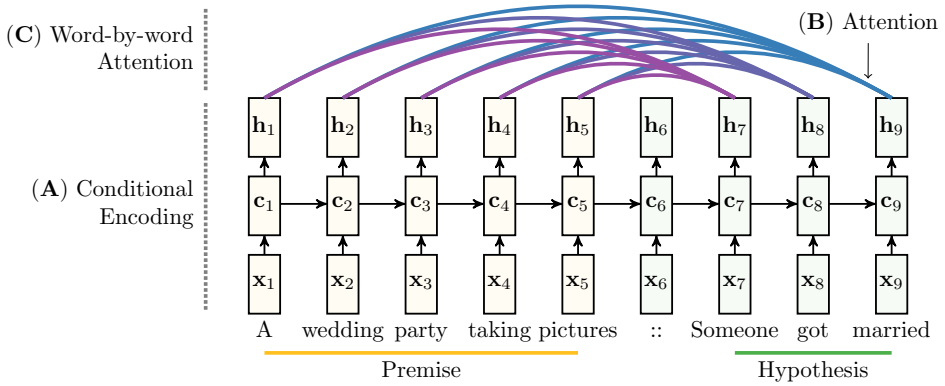


Figure 1: Recognizing textual entailment using (**A**) conditional encoding via two LSTMs, one over the premise and one over the hypothesis conditioned on the representation of the premise ($\mathbf{c}_5$), (**B**) attention only based on the last output vector ($\mathbf{h}_9$) or (**C**) word-by-word attention based on all output vectors of the hypothesis ($\mathbf{h}_7$, $\mathbf{h}_8$ and $\mathbf{h}_9$).

state is initialized with the last cell state of the previous LSTM ($\mathbf{c}_5$ in the example), i.e. it is conditioned on the representation that the first LSTM built for the premise (**A**). We use word2vec vectors (Mikolov et al., 2013) as word representations, which we do not optimize during training. Out-of-vocabulary words in the training set are randomly initialized by sampling values uniformly from $(-0.05, 0.05)$ and optimized during training.[1] Out-of-vocabulary words encountered at inference time on the validation and test corpus are set to fixed random vectors. By not tuning representations of words for which we have word2vec vectors, we ensure that at inference time their representation stays close to unseen similar words for which we have word2vec embeddings. We use a linear layer to project word vectors to the dimensionality of the hidden size of the LSTM, yielding input vectors $\mathbf{x}_i$. Finally, for classification we use a softmax layer over the output of a non-linear projection of the last output vector ($\mathbf{h}_9$ in the example) into the target space of the three classes (ENTAILMENT, NEUTRAL or CONTRADICTION), and train using the cross-entropy loss.

2.3 ATTENTION

Attentive neural networks have recently demonstrated success in a wide range of tasks ranging from handwriting synthesis (Graves, 2013), digit classification (Mnih et al., 2014), machine translation (Bahdanau et al., 2015), image captioning (Xu et al., 2015), speech recognition (Chorowski et al., 2015) and sentence summarization (Rush et al., 2015), to geometric reasoning (Vinyals et al., 2015). The idea is to allow the model to attend over past output vectors (see Figure 1 **B**), thereby mitigating the LSTM's cell state bottleneck. More precisely, an LSTM with attention for RTE does not need to capture the whole semantics of the premise in its cell state. Instead, it is sufficient to output vectors while reading the premise and accumulating a representation in the cell state that informs the second LSTM which of the output vectors of the premise it needs to attend over to determine the RTE class.

Let $\mathbf{Y} \in \mathbb{R}^{k \times L}$ be a matrix consisting of output vectors $[\mathbf{h}_1 \ \cdots \ \mathbf{h}_L]$ that the first LSTM produced when reading the L words of the premise, where k is a hyperparameter denoting the size of embeddings and hidden layers. Furthermore, let $\mathbf{e}_L \in \mathbb{R}^L$ be a vector of 1s and $\mathbf{h}_N$ be the last output vector after the premise and hypothesis were processed by the two LSTMs respectively. The attention mechanism will produce a vector α of attention weights and a weighted representation $\mathbf{r}$ of the premise via

$$\mathbf{M} = \tanh(\mathbf{W}^y\mathbf{Y} + \mathbf{W}^h\mathbf{h}_N \otimes \mathbf{e}_L) \qquad \mathbf{M} \in \mathbb{R}^{k \times L} \tag{7}$$

$$\alpha = \text{softmax}(\mathbf{w}^T\mathbf{M}) \qquad \alpha \in \mathbb{R}^L \tag{8}$$

$$\mathbf{r} = \mathbf{Y}\alpha^T \qquad \mathbf{r} \in \mathbb{R}^k \tag{9}$$

[1] We found 12.1k words in SNLI for which we could not obtain word2vec embeddings, resulting in 3.65M tunable parameters.

where $\mathbf{W}^y, \mathbf{W}^h \in \mathbb{R}^{k \times k}$ are trained projection matrices, $\mathbf{w} \in \mathbb{R}^k$ is a trained parameter vector and $\mathbf{w}^T$ denotes its transpose. Note that the outer product $\mathbf{W}^h \mathbf{h}_N \otimes \mathbf{e}_L$ is repeating the linearly transformed $\mathbf{h}_N$ as many times as there are words in the premise (i.e. L times). Hence, the intermediate attention representation $\mathbf{m}_i$ (ith column vector in $\mathbf{M}$) of the ith word in the premise is obtained from a non-linear combination of the premise's output vector $\mathbf{h}_i$ (ith column vector in $\mathbf{Y}$) and the transformed $\mathbf{h}_N$. The attention weight for the ith word in the premise is the result of a weighted combination (parameterized by $\mathbf{w}$) of values in $\mathbf{m}_i$.

The final sentence-pair representation is obtained from a non-linear combination of the attention-weighted representation $\mathbf{r}$ of the premise and the last output vector $\mathbf{h}_N$ using

$$\mathbf{h}^* = \tanh(\mathbf{W}^p \mathbf{r} + \mathbf{W}^x \mathbf{h}_N) \qquad \mathbf{h}^* \in \mathbb{R}^k \tag{10}$$

where $\mathbf{W}^p, \mathbf{W}^x \in \mathbb{R}^{k \times k}$ are trained projection matrices.

2.4 Word-by-word Attention

For determining whether one sentence entails another it can be a good strategy to check for entailment or contradiction of individual word- and phrase-pairs. To encourage such behavior we employ neural word-by-word attention similar to Bahdanau et al. (2015), Hermann et al. (2015) and Rush et al. (2015). The difference is that we do not use attention to generate words, but to obtain a sentence-pair encoding from fine-grained reasoning via soft-alignment of words and phrases in the premise and hypothesis. In our case, this amounts to attending over the first LSTM's output vectors of the premise while the second LSTM processes the hypothesis one word at a time, thus generating attention weight-vectors α_t over all output vectors of the premise for every word $\mathbf{x}_t$ with $t \in (L+1, N)$ in the hypothesis (Figure 1 **C**). This can be modeled as follows:

$$\mathbf{M}_t = \tanh(\mathbf{W}^y \mathbf{Y} + (\mathbf{W}^h \mathbf{h}_t + \mathbf{W}^r \mathbf{r}_{t-1}) \otimes \mathbf{e}_L) \qquad \mathbf{M}_t \in \mathbb{R}^{k \times L} \tag{11}$$

$$\alpha_t = \text{softmax}(\mathbf{w}^T \mathbf{M}_t) \qquad \alpha_t \in \mathbb{R}^L \tag{12}$$

$$\mathbf{r}_t = \mathbf{Y} \alpha_t^T + \tanh(\mathbf{W}^t \mathbf{r}_{t-1}) \qquad \mathbf{r}_t \in \mathbb{R}^k \tag{13}$$

Note that $\mathbf{r}_t$ is dependent on the previous attention representation $\mathbf{r}_{t-1}$ to inform the model about what was attended over in the previous step (see Eq. 11 and 13).

As in the previous section, the final sentence-pair representation is obtained from a non-linear combination of the last attention-weighted representation of the premise (here based on the last word of the hypothesis) $\mathbf{r}_N$ and the last output vector using

$$\mathbf{h}^* = \tanh(\mathbf{W}^p \mathbf{r}_N + \mathbf{W}^x \mathbf{h}_N) \qquad \mathbf{h}^* \in \mathbb{R}^k \tag{14}$$

2.5 Two-way Attention

Inspired by bidirectional LSTMs that read a sequence and its reverse for improved encoding (Graves and Schmidhuber, 2005), we introduce two-way attention for RTE. The idea is to use the same model (i.e. same structure and weights) to attend over the premise conditioned on the hypothesis, as well as to attend over the hypothesis conditioned on the premise, by simply swapping the two sequences. This produces two sentence-pair representations that we concatenate for classification.

3 Experiments

We conduct experiments on the Stanford Natural Language Inference corpus (SNLI, Bowman et al., 2015). This corpus is two orders of magnitude larger than other existing RTE corpora such as Sentences Involving Compositional Knowledge (SICK, Marelli et al., 2014). Furthermore, a large part of training examples in SICK were generated heuristically from other examples. In contrast, all sentence-pairs in SNLI stem from human annotators. The size and quality of SNLI make it a suitable resource for training neural architectures such as the ones proposed in this paper.

We use ADAM (Kingma and Ba, 2015) for optimization with a first momentum coefficient of 0.9 and a second momentum coefficient of 0.999.[2] For every model we perform a small grid search

[2]Standard configuration recommended by Kingma and Ba.

Table 1: Results on the SNLI corpus.

Model	k	$\lvert\theta\rvert_{\text{W+M}}$	$\lvert\theta\rvert_{\text{M}}$	Train	Dev	Test
Lexicalized classifier (Bowman et al., 2015)	-	-	-	99.7	-	78.2
LSTM (Bowman et al., 2015)	100	$\approx$ 10M	221k	84.4	-	77.6
Conditional encoding, shared	100	3.8M	111k	83.7	81.9	80.9
Conditional encoding, shared	159	3.9M	252k	84.4	83.0	81.4
Conditional encoding	116	3.9M	252k	83.5	82.1	80.9
Attention	100	3.9M	242k	85.4	83.2	82.3
Attention, two-way	100	3.9M	242k	86.5	83.0	82.4
Word-by-word attention	100	3.9M	252k	85.3	**83.7**	**83.5**
Word-by-word attention, two-way	100	3.9M	252k	86.6	83.6	83.2

over combinations of the initial learning rate [1E-4, 3E-4, 1E-3], dropout[3] [0.0, 0.1, 0.2] and ℓ_2-regularization strength [0.0, 1E-4, 3E-4, 1E-3]. Subsequently, we take the best configuration based on performance on the validation set, and evaluate only that configuration on the test set.

4 Results and Discussion

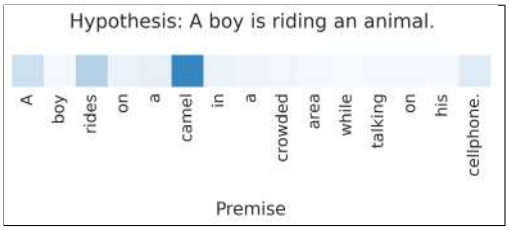
Results on the SNLI corpus are summarized in Table 1. The total number of model parameters, including tunable word representations, is denoted by $\lvert\theta\rvert_{\text{W+M}}$ (without word representations $\lvert\theta\rvert_{\text{M}}$). To ensure a comparable number of parameters to Bowman et al.'s model that encodes the premise and hypothesis independently using one LSTM, we also run experiments for conditional encoding where the parameters of both LSTMs are shared ("Conditional encoding, shared" with $k = 100$) as opposed to using two independent LSTMs. In addition, we compare our attentive models to two benchmark LSTMs whose hidden sizes were chosen so that they have at least as many parameters as the attentive models. Since we are not tuning word vectors for which we have word2vec embeddings, the total number of parameters $\lvert\theta\rvert_{\text{W+M}}$ of our models is considerably smaller. We also compare our models against the benchmark lexicalized classifier used by Bowman et al., which constructs features from the BLEU score between the premise and hypothesis, length difference, word overlap, uni- and bigrams, part-of-speech tags, as well as cross uni- and bigrams.

Conditional Encoding We found that processing the hypothesis conditioned on the premise instead of encoding each sentence independently gives an improvement of 3.3 percentage points in accuracy over Bowman et al.'s LSTM. We argue this is due to information being able to flow from the part of the model that processes the premise to the part that processes the hypothesis. Specifically, the model does not waste capacity on encoding the hypothesis (in fact it does not need to encode the hypothesis at all), but can read the hypothesis in a more focused way by checking words and phrases for contradictions and entailments based on the semantic representation of the premise. One interpretation is that the LSTM is approximating a finite-state automaton for RTE (cf. Angeli and Manning, 2014). Another difference to Bowman et al.'s model is that we are using word2vec instead of GloVe for word representations and, more importantly, do not fine-tune these word embeddings. The drop in accuracy from train to test set is less severe for our models, which suggest that fine-tuning word embeddings could be a cause of overfitting.

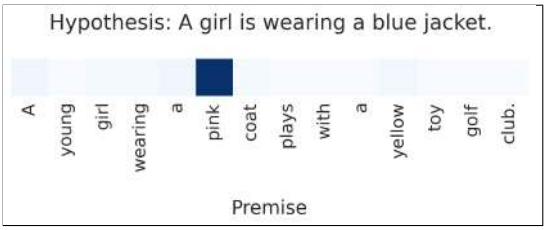
Our LSTM outperforms a simple lexicalized classifier by 2.7 percentage points. To the best of our knowledge, this is the first instance of a neural end-to-end differentiable model to achieve state-of-the-art performance on a textual entailment dataset.

Attention By incorporating an attention mechanism we found a 0.9 percentage point improvement over a single LSTM with a hidden size of 159, and a 1.4 percentage point increase over a benchmark model that uses two LSTMs for conditional encoding (one for the premise and one for the hypothesis conditioned on the representation of the premise). The attention model produces output vectors

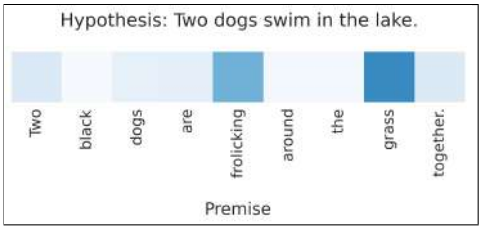
[3] As in Zaremba et al. (2014), we apply dropout only on the inputs and outputs of the network.



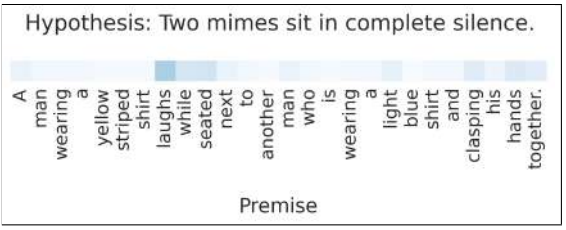
(a)



(b)



(c)



(d)

Figure 2: Attention visualizations.

summarizing contextual information of the premise that is useful to attend over later when reading the hypothesis. Therefore, when reading the premise, the model does not have to build up a semantic representation of the whole premise, but instead a representation that helps attending over the right output vectors when processing the hypothesis. In contrast, the output vectors of the premise are not used by the benchmark LSTMs. Thus, these models have to build up a representation of the whole premise and carry it over through the cell state to the part that processes the hypothesis—a bottleneck that can be overcome to some degree by using attention.

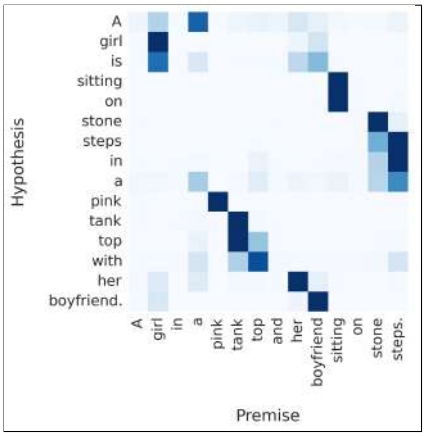
Word-by-word Attention Enabling the model to attend over output vectors of the premise for every word in the hypothesis yields another 1.2 percentage point improvement compared to attending based only on the last output vector of the premise. We argue that this is due to the model being able to check for entailment or contradiction of individual words and phrases in the hypothesis, and demonstrate this effect in the qualitative analysis below.

Two-way Attention Allowing the model to also attend over the hypothesis based on the premise does not seem to improve performance for RTE. We suspect that this is due to entailment being an asymmetric relation. Hence, using the same LSTM to encode the hypothesis (in one direction) and the premise (in the other direction) might lead to noise in the training signal. This could be addressed by training different LSTMs at the cost of doubling the number of model parameters.

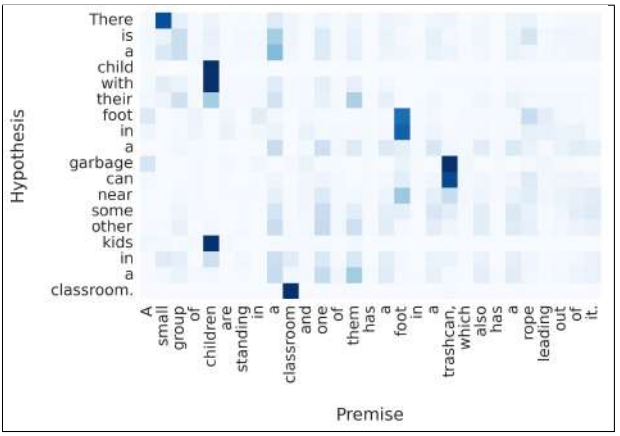
4.1 Qualitative Analysis

It is instructive to analyze which output representations the model is attending over when deciding the class of an RTE example. Note that interpretations based on attention weights have to be taken with care since the model is not forced to solely rely on representations obtained from attention (see $\mathbf{h}_N$ in Eq. 10 and 14). In the following we visualize and discuss the attention patterns of the presented attentive models. For each attentive model we hand-picked examples from ten randomly drawn samples of the validation set.

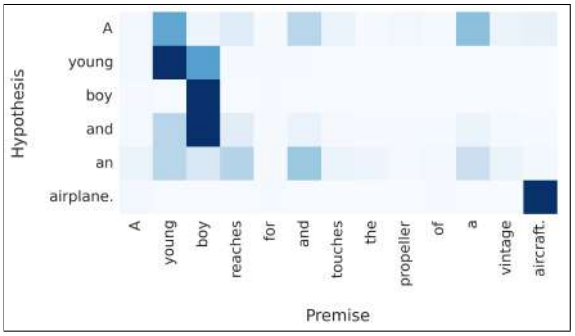
Attention Figure 2 shows to what extent the attentive model focuses on contextual representations of the premise after both LSTMs processed the premise and hypothesis respectively. Note how the model pays attention to output vectors of words that are semantically coherent with the premise ("riding" and "rides", "animal" and "camel", 2a) or in contradiction, as caused by a single word ("blue" vs. "pink", 2b) or multiple words ("swim" and "lake" vs. "frolicking" and "grass", 2c). Interestingly, the model shows contextual understanding by not attending over "yellow", the color



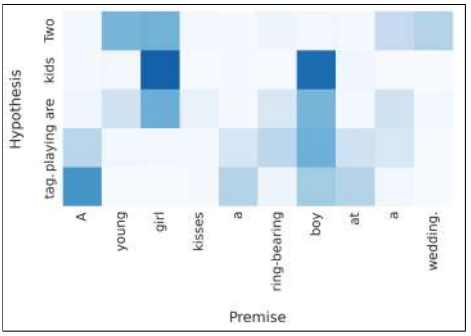
(a)



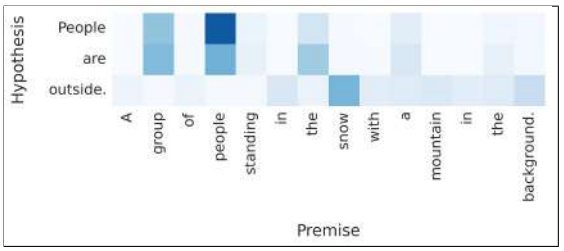
(b)



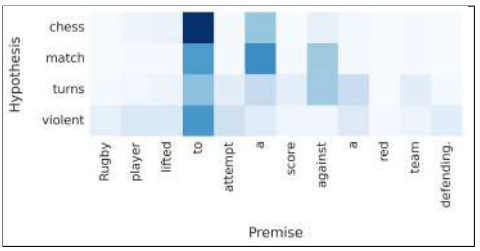
(c)



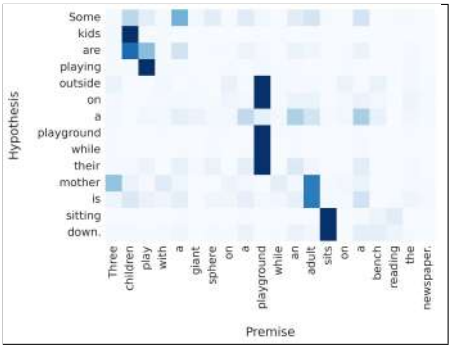
(d)



(e)



(f)



(g)

Figure 3: Word-by-word attention visualizations.

of the toy, but "pink", the color of the coat. However, for more involved examples with longer premises we found that attention is more uniformly distributed (2d). This suggests that conditioning attention only on the last output has limitations when multiple words need to be considered for deciding the RTE class.

Word-by-word Attention Visualizations of word-by-word attention are depicted in Figure 3. We found that word-by-word attention can easily detect if the hypothesis is simply a reordering of words in the premise (3a). Furthermore, it is able to resolve synonyms ("airplane" and "aircraft", 3c) and capable of matching multi-word expressions to single words ("garbage can" to "trashcan", 3b). It is also noteworthy that irrelevant parts of the premise, such as words capturing little meaning or whole uninformative relative clauses, are correctly neglected for determining entailment ("which also has a rope leading out of it", 3b).

Word-by-word attention seems to also work well when words in the premise and hypothesis are connected via deeper semantics or common-sense knowledge ("snow" can be found "outside" and a "mother" is an "adult", 3e and 3g). Furthermore, the model is able to resolve one-to-many relationships ("kids" to "boy" and "girl", 3d)

Attention can fail, for example when the two sentences and their words are entirely unrelated (3f). In such cases, the model seems to back up to attending over function words, and the sentence-pair representation is likely dominated by the last output vector ($\mathbf{h}_N$) instead of the attention-weighted representation (see Eq. 14).

5 Conclusion

In this paper, we show how the state-of-the-art in recognizing textual entailment on a large, human-curated and annotated corpus, can be improved with general end-to-end differentiable models. Our results demonstrate that LSTM recurrent neural networks that read pairs of sequences to produce a final representation from which a simple classifier predicts entailment, outperform both a neural baseline as well as a classifier with hand-engineered features. Extending these models with attention over the premise provides further improvements to the predictive abilities of the system, resulting in a new state-of-the-art accuracy for recognizing entailment on the Stanford Natural Language Inference corpus.

The models presented here are general sequence models, requiring no appeal to Natural Language-specific processing beyond tokenization, and are therefore a suitable target for transfer learning through pre-training the recurrent systems on other corpora, and conversely, applying the models trained on this corpus to other entailment tasks. Future work will focus on such transfer learning tasks, as well as scaling the methods presented here to larger units of text (e.g. paragraphs and entire documents) using hierarchical attention mechanisms. Additionally, it would be worthwhile exploring how other, more structured forms of attention (e.g. Graves et al., 2014; Sukhbaatar et al., 2015), or other forms of differentiable memory (e.g. Grefenstette et al., 2015; Joulin and Mikolov, 2015) could help improve performance on RTE over the neural models presented in this paper. Furthermore, we aim to investigate the application of these generic models to non-natural language sequential entailment problems.

Acknowledgements

We thank Nando de Freitas, Samuel Bowman, Jonathan Berant, Danqi Chen, Christopher Manning, and the anonymous ICLR reviewers for their helpful comments on drafts of this paper.

References

Gabor Angeli and Christopher D Manning. Naturalli: Natural logic inference for common sense reasoning. In *EMNLP*, 2014.

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

Islam Beltagy, Stephen Roller, Pengxiang Cheng, Katrin Erk, and Raymond J. Mooney. Representing meaning with a combination of logical form and vectors. *arXiv preprint arXiv:1505.06816*, 2015.

Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. A large annotated corpus for learning natural language inference. In *EMNLP*, 2015.

Jan Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, Kyunghyun Cho, and Yoshua Bengio. Attention-based models for speech recognition. In *NIPS*, 2015.

Ido Dagan, Oren Glickman, and Bernardo Magnini. The pascal recognising textual entailment challenge. In *Machine Learning Challenges. Lecture Notes in Computer Science*, volume 3944, pages 177–190. Springer, 2006.

Alex Graves. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*, 2013.

Alex Graves and Jürgen Schmidhuber. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural Networks*, 18(5):602–610, 2005.

Alex Graves, Greg Wayne, and Ivo Danihelka. Neural turing machines. *arXiv preprint arXiv:1410.5401*, 2014.

Edward Grefenstette, Karl Moritz Hermann, Mustafa Suleyman, and Phil Blunsom. Learning to transduce with unbounded memory. In *NIPS*, 2015.

Karl Moritz Hermann, Tomáš Kočiský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. Teaching machines to read and comprehend. In *NIPS*, 2015.

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

Baotian Hu, Zhengdong Lu, Hang Li, and Qingcai Chen. Convolutional neural network architectures for matching natural language sentences. In *NIPS*, pages 2042–2050, 2014.

Sergio Jimenez, George Duenas, Julia Baquero, Alexander Gelbukh, Av Juan Dios Bátiz, and Av Mendizábal. Unal-nlp: Combining soft cardinality features for semantic textual similarity, relatedness and entailment. In *SemEval 2014*, page 732, 2014.

Armand Joulin and Tomas Mikolov. Inferring algorithmic patterns with stack-augmented recurrent nets. *arXiv preprint arXiv:1503.01007*, 2015.

Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

Alice Lai and Julia Hockenmaier. Illinois-lh: A denotational and distributional approach to semantics. In *SemEval 2014*, page 329, 2014.

Marco Marelli, Luisa Bentivogli, Marco Baroni, Raffaella Bernardi, Stefano Menini, and Roberto Zamparelli. Semeval-2014 task 1: Evaluation of compositional distributional semantic models on full sentences through semantic relatedness and textual entailment. In *SemEval 2014*, 2014.

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

Volodymyr Mnih, Nicolas Heess, Alex Graves, et al. Recurrent models of visual attention. In *NIPS*, pages 2204–2212, 2014.

Alexander M Rush, Sumit Chopra, and Jason Weston. A neural attention model for abstractive sentence summarization. In *EMNLP*, 2015.

Richard Socher, Eric H Huang, Jeffrey Pennin, Christopher D Manning, and Andrew Y Ng. Dynamic pooling and unfolding recursive autoencoders for paraphrase detection. In *NIPS*, pages 801–809, 2011.

Sainbayar Sukhbaatar, Arthur Szlam, Jason Weston, and Rob Fergus. End-to-end memory networks. *arXiv preprint arXiv:1503.08895*, 2015.

Ilya Sutskever, Oriol Vinyals, and Quoc VV Le. Sequence to sequence learning with neural networks. In *NIPS*, pages 3104–3112, 2014.

Oriol Vinyals, Lukasz Kaiser, Terry Koo, Slav Petrov, Ilya Sutskever, and Geoffrey Hinton. Grammar as a foreign language. In *NIPS*, 2014.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. Pointer networks. In *NIPS*, 2015.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*, 2015.

Wenpeng Yin and Hinrich Schütze. Convolutional neural network for paraphrase identification. In *NAACL-HLT*, pages 901–911, 2015.

Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. Recurrent neural network regularization. *arXiv preprint arXiv:1409.2329*, 2014.

Jiang Zhao, Tian Tian Zhu, and Man Lan. Ecnu: One stone two birds: Ensemble of heterogenous measures for semantic relatedness and textual entailment. In *SemEval 2014*, page 271, 2014.