A Decomposable Attention Model for Natural Language Inference

Ankur P. Parikh
Google
New York, NY

Oscar Täckström
Google
New York, NY

Dipanjan Das
Google
New York, NY

Jakob Uszkoreit
Google
Mountain View, CA

{aparikh,oscart,dipanjand,uszkoreit}@google.com

Abstract

We propose a simple neural architecture for natural language inference. Our approach uses attention to decompose the problem into subproblems that can be solved separately, thus making it trivially parallelizable. On the Stanford Natural Language Inference (SNLI) dataset, we obtain state-of-the-art results with almost an order of magnitude fewer parameters than previous work and without relying on any word-order information. Adding intra-sentence attention that takes a minimum amount of order into account yields further improvements.

1 Introduction

Natural language inference (NLI) refers to the problem of determining entailment and contradiction relationships between a premise and a hypothesis. NLI is a central problem in language understanding (Katz, 1972; Bos and Markert, 2005; van Benthem, 2008; MacCartney and Manning, 2009) and recently the large SNLI corpus of 570K sentence pairs was created for this task (Bowman et al., 2015). We present a new model for NLI and leverage this corpus for comparison with prior work.

A large body of work based on neural networks for text similarity tasks including NLI has been published in recent years (Hu et al., 2014; Rocktäschel et al., 2016; Wang and Jiang, 2016; Yin et al., 2016, *inter alia*). The dominating trend in these models is to build complex, deep text representation models, for example, with convolutional networks (LeCun et al., 1990, CNNs henceforth) or long short-term memory networks (Hochreiter and Schmidhuber, 1997, LSTMs henceforth) with the goal of deeper sentence comprehension. While these approaches have yielded impressive results, they are often computationally very expensive, and result in models having millions of parameters (excluding embeddings).

Here, we take a different approach, arguing that for natural language inference it can often suffice to simply align bits of local text substructure and then aggregate this information. For example, consider the following sentences:

- *Bob is in his room, but because of the thunder and lightning outside, he cannot sleep.*
- *Bob is awake.*
- *It is sunny outside.*

The first sentence is complex in structure and it is challenging to construct a compact representation that expresses its entire meaning. However, it is fairly easy to conclude that the second sentence follows from the first one, by simply aligning *Bob* with *Bob* and *cannot sleep* with *awake* and recognizing that these are synonyms. Similarly, one can conclude that *It is sunny outside* contradicts the first sentence, by aligning *thunder and lightning* with *sunny* and recognizing that these are most likely incompatible.

We leverage this intuition to build a simpler and more lightweight approach to NLI within a neural framework; with considerably fewer parameters, our model outperforms more complex existing neural architectures. In contrast to existing approaches, our approach only relies on alignment and is fully computationally decomposable with respect to the input text. An overview of our approach is given in Figure 1. Given two sentences, where each word is repre-

sented by an embedding vector, we first create a soft alignment matrix using neural attention (Bahdanau et al., 2015). We then use the (soft) alignment to decompose the task into subproblems that are solved separately. Finally, the results of these subproblems are merged to produce the final classification. In addition, we optionally apply intra-sentence attention (Cheng et al., 2016) to endow the model with a richer encoding of substructures prior to the alignment step.

Asymptotically our approach does the same total work as a vanilla LSTM encoder, while being trivially parallelizable across sentence length, which can allow for considerable speedups in low-latency settings. Empirical results on the SNLI corpus show that our approach achieves state-of-the-art results, while using almost an order of magnitude fewer parameters compared to complex LSTM-based approaches.

2 Related Work

Our method is motivated by the central role played by alignment in machine translation (Koehn, 2009) and previous approaches to sentence similarity modeling (Haghighi et al., 2005; Das and Smith, 2009; Chang et al., 2010; Fader et al., 2013), natural language inference (Marsi and Krahmer, 2005; MacCartney et al., 2006; Hickl and Bensley, 2007; MacCartney et al., 2008), and semantic parsing (Andreas et al., 2013). The neural counterpart to alignment, *attention* (Bahdanau et al., 2015), which is a key part of our approach, was originally proposed and has been predominantly used in conjunction with LSTMs (Rocktäschel et al., 2016; Wang and Jiang, 2016) and to a lesser extent with CNNs (Yin et al., 2016). In contrast, our use of attention is purely based on word embeddings and our method essentially consists of feed-forward networks that operate largely independently of word order.

3 Approach

Let $\mathbf{a} = (a_1, \ldots, a_{\ell_a})$ and $\mathbf{b} = (b_1, \ldots, b_{\ell_b})$ be the two input sentences of length ℓ_a and ℓ_b, respectively. We assume that each a_i, $b_j \in \mathbb{R}^d$ is a word embedding vector of dimension d and that each sentence is prepended with a "NULL" token. Our training data comes in the form of labeled pairs $\{\mathbf{a}^{(n)}, \mathbf{b}^{(n)}, \mathbf{y}^{(n)}\}_{n=1}^{N}$, where $\mathbf{y}^{(n)} = (y_1^{(n)}, \ldots, y_C^{(n)})$ is an indicator vector encoding the label and C is the number of output classes. At test time, we receive a pair of sentences $(\mathbf{a}, \mathbf{b})$ and our goal is to predict the correct label $\mathbf{y}$.

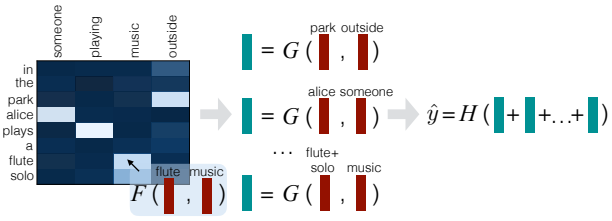

Figure 1: Pictoral overview of the approach, showing the *Attend* (left), *Compare* (center) and *Aggregate* (right) steps.

Input representation. Let $\bar{\mathbf{a}} = (\bar{a}_1, \ldots, \bar{a}_{\ell_a})$ and $\bar{\mathbf{b}} = (\bar{b}_1, \ldots, \bar{b}_{\ell_b})$ denote the input representation of each fragment that is fed to subsequent steps of the algorithm. The vanilla version of our model simply defines $\bar{\mathbf{a}} := \mathbf{a}$ and $\bar{\mathbf{b}} := \mathbf{b}$. With this input representation, our model does not make use of word order. However, we discuss an extension using intra-sentence attention in Section 3.4 that uses a minimal amount of sequence information.

The core model consists of the following three components (see Figure 1), which are trained jointly:

Attend. First, soft-align the elements of $\bar{\mathbf{a}}$ and $\bar{\mathbf{b}}$ using a variant of neural attention (Bahdanau et al., 2015) and decompose the problem into the comparison of aligned subphrases.

Compare. Second, separately compare each aligned subphrase to produce a set of vectors $\{\mathbf{v}_{1,i}\}_{i=1}^{\ell_a}$ for $\mathbf{a}$ and $\{\mathbf{v}_{2,j}\}_{j=1}^{\ell_b}$ for $\mathbf{b}$. Each $\mathbf{v}_{1,i}$ is a nonlinear combination of a_i and its (softly) aligned subphrase in $\mathbf{b}$ (and analogously for $\mathbf{v}_{2,j}$).

Aggregate. Finally, aggregate the sets $\{\mathbf{v}_{1,i}\}_{i=1}^{\ell_a}$ and $\{\mathbf{v}_{2,j}\}_{j=1}^{\ell_b}$ from the previous step and use the result to predict the label $\hat{y}$.

3.1 Attend

We first obtain unnormalized attention weights e_{ij}, computed by a function F', which decomposes as:

$$e_{ij} := F'(\bar{a}_i, \bar{b}_j) := F(\bar{a}_i)^T F(\bar{b}_j)\,. \qquad (1)$$

This decomposition avoids the quadratic complexity that would be associated with separately applying F' $\ell_a \times \ell_b$ times. Instead, only $\ell_a + \ell_b$ applications of F are needed. We take F to be a feed-forward neural network with ReLU activations (Glorot et al., 2011).

These attention weights are normalized as follows:

$$\beta_i := \sum_{j=1}^{\ell_b} \frac{\exp(e_{ij})}{\sum_{k=1}^{\ell_b} \exp(e_{ik})} \bar{b}_j \,,$$

$$\alpha_j := \sum_{i=1}^{\ell_a} \frac{\exp(e_{ij})}{\sum_{k=1}^{\ell_a} \exp(e_{kj})} \bar{a}_i \,. \qquad (2)$$

Here β_i is the subphrase in $\bar{\mathbf{b}}$ that is (softly) aligned to $\bar{a}_i$ and vice versa for α_j.

3.2 Compare

Next, we separately compare the aligned phrases $\{(\bar{a}_i, \beta_i)\}_{i=1}^{\ell_a}$ and $\{(\bar{b}_j, \alpha_j)\}_{j=1}^{\ell_b}$ using a function G, which in this work is again a feed-forward network:

$$\mathbf{v}_{1,i} := G([\bar{a}_i, \beta_i]) \quad \forall i \in [1, \ldots, \ell_a] \,,$$
$$\mathbf{v}_{2,j} := G([\bar{b}_j, \alpha_j]) \quad \forall j \in [1, \ldots, \ell_b] \,. \qquad (3)$$

where the brackets $[\cdot, \cdot]$ denote concatenation. Note that since there are only a linear number of terms in this case, we do not need to apply a decomposition as was done in the previous step. Thus G can jointly take into account both $\bar{a}_i$, and β_i.

3.3 Aggregate

We now have two sets of comparison vectors $\{\mathbf{v}_{1,i}\}_{i=1}^{\ell_a}$ and $\{\mathbf{v}_{2,j}\}_{j=1}^{\ell_b}$. We first aggregate over each set by summation:

$$\mathbf{v}_1 = \sum_{i=1}^{\ell_a} \mathbf{v}_{1,i} \qquad , \qquad \mathbf{v}_2 = \sum_{j=1}^{\ell_b} \mathbf{v}_{2,j} \,. \qquad (4)$$

and feed the result through a final classifier H, that is a feed forward network followed by a linear layer:

$$\hat{\mathbf{y}} = H([\mathbf{v}_1, \mathbf{v}_2]) \,, \qquad (5)$$

where $\hat{\mathbf{y}} \in \mathbb{R}^C$ represents the predicted (unnormalized) scores for each class and consequently the predicted class is given by $\hat{y} = \text{argmax}_i \hat{\mathbf{y}}_i$.

For training, we use multi-class cross-entropy loss with dropout regularization (Srivastava et al., 2014):

$$L(\theta_F, \theta_G, \theta_H) = \frac{1}{N} \sum_{n=1}^{N} \sum_{c=1}^{C} y_c^{(n)} \log \frac{\exp(\hat{y}_c)}{\sum_{c'=1}^{C} \exp(\hat{y}_{c'})} \,.$$

Here $\theta_F, \theta_G, \theta_H$ denote the learnable parameters of the functions F, G and H, respectively.

3.4 Intra-Sentence Attention (Optional)

In the above model, the input representations are simple word embeddings. However, we can augment this input representation with *intra-sentence attention* to encode compositional relationships between words within each sentence, as proposed by Cheng et al. (2016). Similar to Eqs. 1 and 2, we define

$$f_{ij} := F_{\text{intra}}(a_i)^T F_{\text{intra}}(a_j) \,, \qquad (6)$$

where F_{intra} is a feed-forward network. We then create the self-aligned phrases

$$a'_i := \sum_{j=1}^{\ell_a} \frac{\exp(f_{ij} + d_{i-j})}{\sum_{k=1}^{\ell_a} \exp(f_{ik} + d_{i-k})} a_j \,. \qquad (7)$$

The distance-sensitive bias terms $d_{i-j} \in \mathbb{R}$ provides the model with a minimal amount of sequence information, while remaining parallelizable. These terms are bucketed such that all distances greater than 10 words share the same bias. The input representation for subsequent steps is then defined as $\bar{a}_i := [a_i, a'_i]$ and analogously $\bar{b}_i := [b_i, b'_i]$.

4 Computational Complexity

We now discuss the asymptotic complexity of our approach and how it offers a higher degree of parallelism than LSTM-based approaches. Recall that d denotes embedding dimension and ℓ means sentence length. For simplicity we assume that all hidden dimensions are d and that the complexity of matrix($d \times d$)-vector($d \times 1$) multiplication is $O(d^2)$.

A key assumption of our analysis is that $\ell < d$, which we believe is reasonable and is true of the SNLI dataset (Bowman et al., 2015) where $\ell < 80$, whereas recent LSTM-based approaches have used $d \geq 300$. This assumption allows us to bound the complexity of computing the ℓ^2 attention weights.

Complexity of LSTMs. The complexity of an LSTM cell is $O(d^2)$, resulting in a complexity of $O(\ell d^2)$ to encode the sentence. Adding attention as in Rocktäschel et al. (2016) increases this complexity to $O(\ell d^2 + \ell^2 d)$.

Complexity of our Approach. Application of a feed-forward network requires $O(d^2)$ steps. Thus, the **Compare** and **Aggregate** steps have complexity $O(\ell d^2)$ and $O(d^2)$ respectively. For the **Attend** step,

Method	Train Acc	Test Acc	#Parameters
Lexicalized Classifier (Bowman et al., 2015)	99.7	78.2	–
300D LSTM RNN encoders (Bowman et al., 2016)	83.9	80.6	3.0M
1024D pretrained GRU encoders (Vendrov et al., 2015)	98.8	81.4	15.0M
300D tree-based CNN encoders (Mou et al., 2015)	83.3	82.1	3.5M
300D SPINN-PI encoders (Bowman et al., 2016)	89.2	83.2	3.7M
100D LSTM with attention (Rocktäschel et al., 2016)	85.3	83.5	252K
300D mLSTM (Wang and Jiang, 2016)	92.0	86.1	1.9M
450D LSTMN with deep attention fusion (Cheng et al., 2016)	88.5	86.3	3.4M
Our approach (vanilla)	89.5	86.3	382K
Our approach with intra-sentence attention	90.5	**86.8**	582K

Table 1: Train/test accuracies on the SNLI dataset and number of parameters (excluding embeddings) for each approach.

Method	**N**	**E**	**C**
Bowman et al. (2016)	80.6	88.2	85.5
Wang and Jiang (2016)	81.6	91.6	87.4
Our approach (vanilla)	83.6	91.3	85.8
Our approach w/ intra att.	83.7	92.1	86.7

Table 2: Breakdown of accuracy with respect to classes on SNLI development set. **N**=neutral, **E**=entailment, **C**=contradiction.

F is evaluated $O(\ell)$ times, giving a complexity of $O(\ell d^2)$. Each attention weight e_{ij} requires one dot product, resulting in a complexity of $O(\ell^2 d)$.

Thus the total complexity of the model is $O(\ell d^2 + \ell^2 d)$, which is equal to that of an LSTM with attention. However, note that with the assumption that $\ell < d$, this becomes $O(\ell d^2)$ which is the same complexity as a regular LSTM. Moreover, unlike the LSTM, our approach has the advantage of being parallelizable over ℓ, which can be useful at test time.

5 Experiments

We evaluate our approach on the Stanford Natural Language Inference (SNLI) dataset (Bowman et al., 2015). Given a sentences pair $(\mathbf{a}, \mathbf{b})$, the task is to predict whether $\mathbf{b}$ is *entailed* by $\mathbf{a}$, $\mathbf{b}$ *contradicts* $\mathbf{a}$, or whether their relationship is *neutral*.

5.1 Implementation Details

The method was implemented in TensorFlow (Abadi et al., 2015).

Data preprocessing: Following Bowman et al. (2015), we remove examples labeled "–" (no gold label) from the dataset, which leaves 549,367 pairs for training, 9,842 for development, and 9,824 for testing. We use the tokenized sentences from the non-binary parse provided in the dataset and prepend each sentence with a "NULL" token. During training, each sentence was padded up to the maximum length of the batch for efficient training (the padding was explicitly masked out so as not to affect the objective/gradients). For efficient batching in TensorFlow, we semi-sorted the training data to first contain examples where both sentences had length less than 20, followed by those with length less than 50, and then the rest. This ensured that most training batches contained examples of similar length.

Embeddings: We use 300 dimensional GloVe embeddings (Pennington et al., 2014) to represent words. Each embedding vector was normalized to have ℓ_2 norm of 1 and projected down to 200 dimensions, a number determined via hyperparameter tuning. Out-of-vocabulary (OOV) words are hashed to one of 100 random embeddings each initialized to mean 0 and standard deviation 1. All embeddings remain fixed during training, but the projection matrix is trained. All other parameter weights (hidden layers etc.) were initialized from random Gaussians with mean 0 and standard deviation 0.01.

Each hyperparameter setting was run on a single machine with 10 asynchronous gradient-update threads, using Adagrad (Duchi et al., 2011) for optimization with the default initial accumulator value of 0.1. Dropout regularization (Srivastava et al., 2014) was used for all ReLU layers, but not for the final linear layer. We additionally tuned the following hyperparameters and present their chosen values in

ID	Sentence 1	Sentence 2	DA (vanilla)	DA (intra att.)	SPINN-PI	mLSTM	Gold
A	Two kids are standing in the ocean hugging each other.	Two kids enjoy their day at the beach.	**N**	**N**	E	E	**N**
B	A dancer in costumer performs on stage while a man watches.	the man is captivated	**N**	**N**	E	E	**N**
C	They are sitting on the edge of a fountain	The fountain is splashing the persons seated.	**N**	**N**	C	C	**N**
D	Two dogs play with tennis ball in field.	Dogs are watching a tennis match.	N	**C**	**C**	**C**	**C**
E	Two kids begin to make a snowman on a sunny winter day.	Two penguins making a snowman.	N	**C**	**C**	**C**	**C**
F	The horses pull the carriage, holding people and a dog, through the rain.	Horses ride in a carriage pulled by a dog.	E	E	**C**	**C**	**C**
G	A woman closes her eyes as she plays her cello.	The woman has her eyes open.	E	E	E	E	**C**
H	Two women having drinks and smoking cigarettes at the bar.	Three women are at a bar.	E	E	E	E	**C**
I	A band playing with fans watching.	A band watches the fans play	E	E	E	E	**C**

Table 3: Example wins and losses compared to other approaches. DA (Decomposable Attention) refers to our approach while SPINN-PI and mLSTM are previously developed methods (see Table 1).

parentheses: network size (2-layers, each with 200 neurons), batch size (4), [1] dropout ratio (0.2) and learning rate (0.05–vanilla, 0.025–intra-attention). All settings were run for 50 million steps (each step indicates one batch) but model parameters were saved frequently as training progressed and we chose the model that did best on the development set.

5.2 Results

Results in terms of 3-class accuracy are shown in Table 1. Our vanilla approach achieves state-of-the-art results with almost an order of magnitude fewer parameters than the LSTMN of Cheng et al. (2016). Adding intra-sentence attention gives a considerable improvement of 0.5 percentage points over the existing state of the art. Table 2 gives a breakdown of accuracy on the development set showing that most of our gains stem from *neutral*, while most losses come from *contradiction* pairs.

Table 3 shows some wins and losses. Examples A-C are cases where both variants of our approach are correct while both SPINN-PI (Bowman et al., 2016) and the mLSTM (Wang and Jiang, 2016) are incorrect. In the first two cases, both sentences contain phrases that are either identical or highly lexically related (e.g. "Two kids" and "ocean / beach") and our approach correctly favors neutral in these cases. In Example C, it is possible that relying on word-order may confuse SPINN-PI and the mLSTM due to how "fountain" is the object of a preposition in the first sentence but the subject of the second.

The second set of examples (D-F) are cases where our vanilla approach is incorrect but mLSTM and SPINN-PI are correct. Example F requires sequential information and neither variant of our approach can predict the correct class. Examples D-E are interesting however, since they don't require word order information, yet intra-attention seems to help. We suspect this may be because the word embeddings are not fine-grained enough for the algorithm to conclude that "play/watch" is a contradiction, but intra-attention, by adding an extra layer of composition/nonlinearity to incorporate context, compensates for this.

Finally, Examples G-I are cases that all methods get wrong. The first is actually representative of many examples in this category where there is one critical word that separates the two sentences (close vs open in this case) and goes unnoticed by the algorithms. Examples H requires inference about numbers and Example I needs sequence information.

6 Conclusion

We presented a simple attention-based approach to natural language inference that is trivially parallelizable. The approach outperforms considerably more complex neural methods aiming for text understanding. Our results suggest that, at least for this task, pairwise comparisons are relatively more important than global sentence-level representations.

Acknowledgements

We thank Slav Petrov, Tom Kwiatkowski, Yoon Kim, Erick Fonseca, Mark Neumann for useful discussion and Sam Bowman and Shuohang Wang for providing us their model outputs for error analysis.

[1] 16 or 32 also work well and are a bit more stable.

References

[Abadi et al.2015] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. 2015. TensorFlow: Large-scale machine learning on heterogeneous systems. *Software available from tensorflow. org*.

[Andreas et al.2013] Jacob Andreas, Andreas Vlachos, and Stephen Clark. 2013. Semantic parsing as machine translation. In *Proceedings of ACL*.

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

[Bos and Markert2005] Johan Bos and Katja Markert. 2005. Recognising textual entailment with logical inference. In *Proceedings of EMNLP*.

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

[Bowman et al.2016] 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 ACL*.

[Chang et al.2010] Ming-Wei Chang, Dan Goldwasser, Dan Roth, and Vivek Srikumar. 2010. Discriminative learning over constrained latent representations. In *Proceedings of HLT-NAACL*.

[Cheng et al.2016] Jianpeng Cheng, Li Dong, and Mirella Lapata. 2016. Long short-term memory-networks for machine reading. In *Proceedings of EMNLP*.

[Das and Smith2009] Dipanjan Das and Noah A. Smith. 2009. Paraphrase identification as probabilistic quasi-synchronous recognition. In *Proceedings of ACL-IJCNLP*.

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

[Fader et al.2013] Anthony Fader, Luke S Zettlemoyer, and Oren Etzioni. 2013. Paraphrase-driven learning for open question answering. In *Proceedings of ACL*.

[Glorot et al.2011] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. 2011. Deep sparse rectifier neural networks. In *Proceedings of AISTATS*.

[Haghighi et al.2005] Aria D. Haghighi, Andrew Y. Ng, and Christopher D. Manning. 2005. Robust textual inference via graph matching. In *Proceedings of HLT-NAACL*.

[Hickl and Bensley2007] Andrew Hickl and Jeremy Bensley. 2007. A discourse commitment-based framework for recognizing textual entailment. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*. Association for Computational Linguistics.

[Hochreiter and Schmidhuber1997] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

[Hu et al.2014] Baotian Hu, Zhengdong Lu, Hang Li, and Qingcai Chen. 2014. Convolutional neural network architectures for matching natural language sentences. In *Advances in NIPS*.

[Katz1972] Jerrold J. Katz. 1972. *Semantic theory*. Harper & Row.

[Koehn2009] Philipp Koehn. 2009. *Statistical machine translation*. Cambridge University Press.

[LeCun et al.1990] Y. LeCun, B. Boser, J.S. Denker, D. Henderson, R.E. Howard, W. Hubbard, and L.D. Jackel. 1990. Handwritten digit recognition with a back-propagation network. In *Advances in NIPS*.

[MacCartney and Manning2009] Bill MacCartney and Christopher D. Manning. 2009. An extended model of natural logic. In *Proceedings of the IWCS*.

[MacCartney et al.2006] Bill MacCartney, Trond Grenager, Marie-Catherine de Marneffe, Daniel Cer, and Christopher D Manning. 2006. Learning to recognize features of valid textual entailments. In *Proceedings of HLT-NAACL*.

[MacCartney et al.2008] Bill MacCartney, Michel Galley, and Christopher D Manning. 2008. A phrase-based alignment model for natural language inference. In *Proceedings of EMNLP*.

[Marsi and Krahmer2005] Erwin Marsi and Emiel Krahmer. 2005. Classification of semantic relations by humans and machines. In *Proceedings of the ACL workshop on Empirical Modeling of Semantic Equivalence and Entailment*.

[Mou et al.2015] Lili Mou, Men Rui, Ge Li, Yan Xu, Lu Zhang, Rui Yan, and Zhi Jin. 2015. Natural language inference by tree-based convolution and heuristic matching. In *Proceedings of ACL (short papers)*.

[Pennington et al.2014] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of EMNLP*.

[Rocktäschel et al.2016] Tim Rocktäschel, Edward Grefenstette, Karl Moritz Hermann, Tomáš Kočiskỳ, and Phil Blunsom. 2016. Reasoning about entailment with neural attention. In *Proceedings of ICLR*.

[Srivastava et al.2014] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958.

[van Benthem2008] Johan van Benthem. 2008. *A brief history of natural logic*. College Publications.

[Vendrov et al.2015] Ivan Vendrov, Ryan Kiros, Sanja Fidler, and Raquel Urtasun. 2015. Order-embeddings of images and language. In *Proceedings of ICLR*.

[Wang and Jiang2016] Shuohang Wang and Jing Jiang. 2016. Learning natural language inference with LSTM. In *Proceedings of NAACL*.

[Yin et al.2016] Wenpeng Yin, Hinrich Schütze, Bing Xiang, and Bowen Zhou. 2016. ABCNN: Attention-based convolutional neural network for modeling sentence pairs. In *Transactions of the Association of Computational Linguistics*.