Generating Natural Language Inference Chains

Vladyslav Kolesnyk and **Tim Rocktäschel** and **Sebastian Riedel**
University College London
London, UK
vladyslav.kolesnyk.12@ucl.ac.uk, {t.rocktaschel,s.riedel}@cs.ucl.ac.uk

Abstract

The ability to reason with natural language is a fundamental prerequisite for many NLP tasks such as information extraction, machine translation and question answering. To quantify this ability, systems are commonly tested whether they can recognize textual entailment, *i.e.*, whether one sentence can be inferred from another one. However, in most NLP applications only single source sentences instead of sentence pairs are available. Hence, we propose a new task that measures how well a model can *generate* an entailed sentence from a source sentence. We take entailment-pairs of the Stanford Natural Language Inference corpus and train an LSTM with attention. On a manually annotated test set we found that 82% of generated sentences are correct, an improvement of 10.3% over an LSTM baseline. A qualitative analysis shows that this model is not only capable of shortening input sentences, but also inferring new statements via paraphrasing and phrase entailment. We then apply this model recursively to input-output pairs, thereby generating natural language inference chains that can be used to automatically construct an entailment graph from source sentences. Finally, by swapping source and target sentences we can also train a model that given an input sentence invents additional information to generate a new sentence.

1 Introduction

The ability to determine entailment or contradiction between natural language text is essential for improving the performance in a wide range of natural language processing tasks. Recognizing Textual Entailment (RTE) is a task primarily designed to determine whether two natural language sentences are independent, contradictory or in an entailment relationship where the second sentence (the hypothesis) can be inferred from the first (the premise). Although systems that perform well in RTE could potentially be used to improve question answering, information extraction, text summarization and machine translation [6], only in few of such downstream NLP tasks sentence-pairs are actually available. Usually, only a single source sentence (*e.g.* a question that needs to be answered or a source sentence that we want to translate) is present and models need to come up with their own hypotheses and commonsense knowledge inferences.

The release of the large Stanford Natural Language Inference (SNLI) corpus [2] allowed end-to-end differentiable neural networks to outperform feature-based classifiers on the RTE task [3, 4, 12, 15, 17].

In this work, we go a step further and investigate how well recurrent neural networks can produce true hypotheses given a source sentence. Furthermore, we qualitatively demonstrate that by only training on input-output pairs and recursively generating entailed sentence we can generate natural language inference chains (see Figure 1 for an example). Note that every inference step is interpretable as it is mapping one natural language sentence to another one.

Our contributions are fourfold: (i) we propose

Young blond woman putting her foot into a water fountain → A person is dipping her foot into water. → A person is wet.

Figure 1: Example of a natural language inference chain that is generated from the first sentence.

an entailment generation task based on the SNLI corpus (§2.1), (ii) we investigate a sequence-to-sequence model and find that 82% of generated sentences are correct (§3.2), (iii) we demonstrate the ability to generate natural language inference chains trained solely from entailment pairs (§3.3), and finally (iv) we can also generate sentences with more specific information by swapping source and target sentences during training (§3.4).

2 Method

In the section, we briefly introduce the entailment generation task and our sequence-to-sequence model.

2.1 Entailment Generation

To create the entailment generation dataset, we simply filter the Stanford Natural Language Inference corpus for sentence-pairs of the entailment class. This results in a training set of $183,416$ sentence pairs, a development set of $3,329$ pairs and a test of $3,368$ pairs. Instead of a classification task, we can now use this dataset for a sequence transduction task.

2.2 Sequence-to-Sequence

Sequence-to-sequence recurrent neural networks [14] have been successfully employed for many sequence transduction tasks in NLP such as machine translation [1, 5], constituency parsing [16], sentence summarization [13] and question answering [7]. They consist of two recurrent neural networks (RNNs): an encoder that maps an input sequence of words into a dense vector representation, and a decoder that conditioned on that vector representation generates an output sequence. Specifically, we use long short-term memory (LSTM) RNNs [8] for encoding and decoding. Furthermore, we experiment with *word-by-word attention* [1], which allows the decoder to search in the encoder outputs to circumvent the LSTM's memory bottleneck. We use greedy decoding at test time. The success of LSTMs with attention in sequence transduction tasks makes them a natural choice as a baseline for entailment generation, and we leave the investigation of more advanced models to future work.

2.3 Optimization and Hyperparameters

We use stochastic gradient descent with a mini-batch size of 64 and the ADAM optimizer [9] with a first momentum coefficient of 0.9 and a second momentum coefficient of 0.999. Word embeddings are initialized with pre-trained word2vec vectors [10]. Out-of-vocabulary words (10.5%) are randomly initialized by sampling values uniformly from $[-\sqrt{3}, \sqrt{3}]$ and optimized during training. Furthermore, we clip gradients using a norm of 5.0. We stop training after 25 epochs.

3 Experiments and Results

We present results for various tasks: (i) given a premise, generate a sentence that can be inferred from the premise, (ii) construct inference chains by recursively generating sentences, and (iii) given a sentence, create a premise that would entail this sentence, *i.e.*, make a more descriptive sentence by adding specific information.

3.1 Quantitative Evaluation

We train an LSTM with and without attention on the training set. After training, we take the best model in terms of BLEU score [11] on the development set and calculate the BLEU score on the test set. To our surprise, we found that using attention yields only a marginally higher BLEU score (43.1 vs. 42.8). We suspect that this is due to the fact that generating entailed sentences has a larger space of valid target sequences, which makes the use of BLEU problematic and penalizes correct solutions. Hence, we manually annotated 100 random test sentences and decided whether the generated sentence can indeed be inferred from the source sentence. We found that sentences generated by an LSTM with attention are substantially more accurate (82% accuracy) than those generated from an LSTM baseline (71.7%). To gain more insights into the model's capabilities, we turn to a thorough qualitative

1. He is driving a beautiful red car. → A man is driving a car.
2. Two men and a woman finishing a meal and drinks. → Three people are eating together.
3. A father holds a new born baby. → A man holds his baby.

Figure 2: Valid sentences generated by the model.

1. A female gymnast in black and red being coached on bar skills. → A female is in a bar.
2. A dog drops a yellow disc on a beach. → The dog drops a red disc.

Figure 3: Invalid sentences generated by the model.

analysis of the attention LSTM model in the remainder of this paper.

3.2 Example Generations

Figure 2 shows examples of generated sentences from the development set. Syntactic simplification of the input sentence seems to be the most common approach. The model removes certain parts of the premise such as adjectives, resulting in a more abstract sentence (see Figure 2.1).

Figure 2.2 demonstrates that the system can recognize the number of subjects in the sentence and includes this information in the generated sentence. However, we did not observe such 'counting' behavior for more than four subjects, indicating that the system memorized frequency patterns from the training set.

Furthermore, we found predictions that hint to common-sense assumptions: if a sentence talks about a father holding a newborn baby, it is most likely that the newborn baby is his own child (Example 2.3).

Limitations Two reappearing limitations of the proposed model are related to dealing with words that have a very different meaning but similar word2vec embeddings (*e.g.* colors), as well as ambiguous words. For instance, 'bar' in Figure 3.1 refers to pole vault and not a place in which you can have a drink. Substituting one color by another one (Figure 3.2) is a common mistake.

1. First time a computer Go program has beaten a 9-dan professional without handicaps. → A person is using a computer.
2. Competitors must be dressed in suitable tennis attire that is almost entirely white. → The tennis player is wearing white.
3. The athletes run by the harbour during the Haspa Hamburg marathon. → The runners are outside.
4. Space travellers gaze up and strain to find the blue dot in their sky. → There are people looking at the sky.

Figure 4: Generated sentences from external sources: (1) Wikipedia, (2) wimbledon.com, (3) a news article, (4) "The Frontier is Everywhere" by Carl Sagan.

Out-of-Corpus Examples The SNLI corpus might not reflect the variety of sentences that can be encountered in downstream NLP tasks. In Figure 4 we present generated sentences for randomly selected examples of out-of-domain textual resources. They demonstrate that the model generalizes well to out-of-domain sentences, making it a potentially very useful component for improving systems for question answering, information extraction, sentence summarization etc.

3.3 Inference Chain Generation

Next, we test how well the model can generate inference chains by repeatedly passing generated output sentences as inputs to the model. We stop once a sentence has already been generated in the chain. Figure 5 shows that this works well despite that the model was only trained on sentence-pairs.

Furthermore, by generating inference chains for all sentences in the development set we construct an entailment graph. In that graph we found that sentences with shared semantics are eventually mapped to the same sentence that captures the shared meaning.

1. People are celebrating a victory on the square. → People are celebrating. → People are happy. → The people are smiling. ← People are smiling. ← Some people smiling. ← couple smiling ← A bride and groom smiles. ← A wedding party looks happy on the picture.

2. The athletes run at the start of the Haspa Hamburg marathon. → The runners are at the race. → There are runners at the race. → There are people running. → People are running. ← People are running outside. ← People are running away from the bear.

Figure 5: Examples of inference chains where two premises (underlined) converge to the same sentence (highlighted).

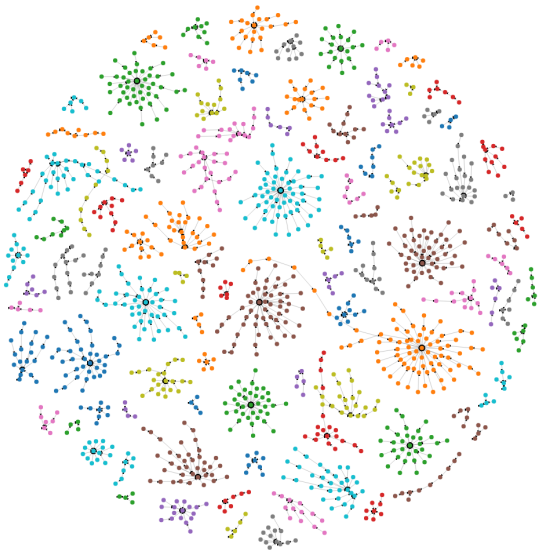

Figure 6: Topology of the generated entailment graph.

A visualization of the topology of the entailment graph is shown in Figure 6. Note that there are several long inference chains, as well as large clusters of sentences (nodes) that are mapped (links) to the same shared meaning.

3.4 Inverse Inference

By swapping the source and target sequences for training, we can train a model that given a sentence invents additional information to generate a new sentence (Figure 7). We believe this might prove useful to increase the language variety and

1. Two girls sit by a lake. → Two girls are sitting on a ledge in front of a lake.

2. Two girls are playing chess → Two girls playing chess in a restaurant.

3. A woman is outside. → A woman is walking down a street in a city. → A woman in a black skirt walks down a city street.

Figure 7: Generated sentences for the inverse task.

complexity of AI unit tests such as the Facebook bAbI task [18], but we leave this for future work.

4 Conclusion and Future Work

We investigated the ability of sequence-to-sequence models to generate entailed sentences from a source sentence. To this end, we trained an attentive LSTM on entailment-pairs of the SNLI corpus. We found that this works well and generalizes beyond in-domain sentences. Hence, it could become a useful component for improving the performance of other NLP systems.

We were able to generate natural language inference chains by recursively generating sentences from previously inferred ones. This allowed us to construct an entailment graph for sentences of the SNLI development corpus. In this graph, the shared meaning of two related sentences is represented by the first natural language sentence that connects both sentences. Every inference step is interpretable as it maps a natural language sentence to another one.

Towards high-quality data augmentation, we experimented with reversing the generation task. We found that this enabled the model to learn to invent specific information.

For future work, we want to integrate the presented model into larger architectures to improve the performance of downstream NLP tasks such as information extraction and question answering. Furthermore, we plan to use the model for data augmentation to train expressive neural networks on tasks where only little annotated data is available. Another interesting research direction is to investigate methods for increasing the diversity of the generated sentences.

Acknowledgments

We thank Guillaume Bouchard for suggesting the reversed generation task, and Dirk Weissenborn, Isabelle Augenstein and Matko Bosnjak for comments on drafts of this paper. This work was supported by Microsoft Research through its PhD Scholarship Programme, an Allen Distinguished Investigator Award, and a Marie Curie Career Integration Award.

References

[1] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. In *ICLR*, 2015.

[2] S. R. Bowman, G. Angeli, C. Potts, and C. D. Manning. A large annotated corpus for learning natural language inference. In *EMNLP*, 2015.

[3] S. R. Bowman, J. Gauthier, A. Rastogi, R. Gupta, C. D. Manning, and C. Potts. A fast unified model for parsing and sentence understanding. In *ACL*, 2016.

[4] J. Cheng, L. Dong, and M. Lapata. Long short-term memory-networks for machine reading. *arXiv preprint arXiv:1601.06733*, 2016.

[5] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[6] I. Dagan, O. Glickman, and B. Magnini. The pascal recognising textual entailment challenge. In *Machine learning challenges. evaluating predictive uncertainty, visual object classification, and recognising tectual entailment*, pages 177–190. Springer, 2006.

[7] K. M. Hermann, T. Kocisky, E. Grefenstette, L. Espeholt, W. Kay, M. Suleyman, and P. Blunsom. Teaching machines to read and comprehend. In *NIPS*, pages 1684–1692, 2015.

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

[9] D. Kingma and J. Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[10] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *NIPS*, pages 3111–3119, 2013.

[11] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. Bleu: a method for automatic evaluation of machine translation. In *ACL*, pages 311–318, 2002.

[12] T. Rocktäschel, E. Grefenstette, K. M. Hermann, T. Kočiskỳ, and P. Blunsom. Reasoning about entailment with neural attention. In *ICLR*, 2016.

[13] A. M. Rush, S. Chopra, and J. Weston. A neural attention model for abstractive sentence summarization. In *EMNLP*, 2015.

[14] I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. In *NIPS*, pages 3104–3112, 2014.

[15] I. Vendrov, R. Kiros, S. Fidler, and R. Urtasun. Order-embeddings of images and language. In *ICLR*, 2016.

[16] O. Vinyals, Ł. Kaiser, T. Koo, S. Petrov, I. Sutskever, and G. Hinton. Grammar as a foreign language. In *NIPS*, pages 2755–2763, 2015.

[17] S. Wang and J. Jiang. Learning Natural Language Inference with LSTM. In *NAACL*, 2016.

[18] J. Weston, A. Bordes, S. Chopra, and T. Mikolov. Towards ai-complete question answering: A set of prerequisite toy tasks. In *ICLR*, 2016.