Neural Open Information Extraction

Lei Cui, **Furu Wei**, and **Ming Zhou**

Microsoft Research Asia
{lecu,fuwei,mingzhou}@microsoft.com

Abstract

Conventional Open Information Extraction (Open IE) systems are usually built on hand-crafted patterns from other NLP tools such as syntactic parsing, yet they face problems of error propagation. In this paper, we propose a neural Open IE approach with an encoder-decoder framework. Distinct from existing methods, the neural Open IE approach learns highly confident arguments and relation tuples bootstrapped from a state-of-the-art Open IE system. An empirical study on a large benchmark dataset shows that the neural Open IE system significantly outperforms several baselines, while maintaining comparable computational efficiency.

1 Introduction

Open Information Extraction (Open IE) involves generating a structured representation of information in text, usually in the form of triples or n-ary propositions. An Open IE system not only extracts arguments but also relation phrases from the given text, which does not rely on pre-defined ontology schema. For instance, given the sentence "*deep learning is a subfield of machine learning*", the triple (*deep learning*; *is a subfield of*; *machine learning*) can be extracted, where the relation phrase "*is a subfield of*" indicates the semantic relationship between two arguments. Open IE plays a key role in natural language understanding and fosters many downstream NLP applications such as knowledge base construction, question answering, text comprehension, and others.

The Open IE system was first introduced by TEXTRUNNER (Banko et al., 2007), followed by several popular systems such as REVERB (Fader et al., 2011), OLLIE (Mausam et al., 2012), ClausIE (Del Corro and Gemulla, 2013) Stanford OPENIE (Angeli et al., 2015), PropS (Stanovsky et al., 2016) and most recently OPENIE4[1] (Mausam, 2016) and OPENIE5[2]. Although these systems have been widely used in a variety of applications, most of them were built on hand-crafted patterns from syntactic parsing, which causes errors in propagation and compounding at each stage (Banko et al., 2007; Gashteovski et al., 2017; Schneider et al., 2017). Therefore, it is essential to solve the problems of cascading errors to alleviate extracting incorrect tuples.

To this end, we propose a neural Open IE approach with an encoder-decoder framework. The encoder-decoder framework is a text generation technique and has been successfully applied to many tasks, such as machine translation (Cho et al., 2014; Bahdanau et al., 2014; Sutskever et al., 2014; Wu et al., 2016; Gehring et al., 2017; Vaswani et al., 2017), image caption (Vinyals et al., 2014), abstractive summarization (Rush et al., 2015; Nallapati et al., 2016; See et al., 2017) and recently keyphrase extraction (Meng et al., 2017). Generally, the encoder encodes the input sequence to an internal representation called 'context vector' which is used by the decoder to generate the output sequence. The lengths of input and output sequences can be different, as there is no one on one relation between the input and output sequences. In this work, Open IE is cast as a sequence-to-sequence generation problem, where the input sequence is the sentence and the output sequence is the tuples with special placeholders. For instance, given the input sequence "*deep learning is a subfield of machine learning*", the output sequence will be "⟨arg1⟩ deep learning ⟨/arg1⟩ ⟨rel⟩ is a subfield of ⟨/rel⟩ ⟨arg2⟩ machine

[1] https://github.com/allenai/openie-standalone
[2] https://github.com/dair-iitd/OpenIE-standalone

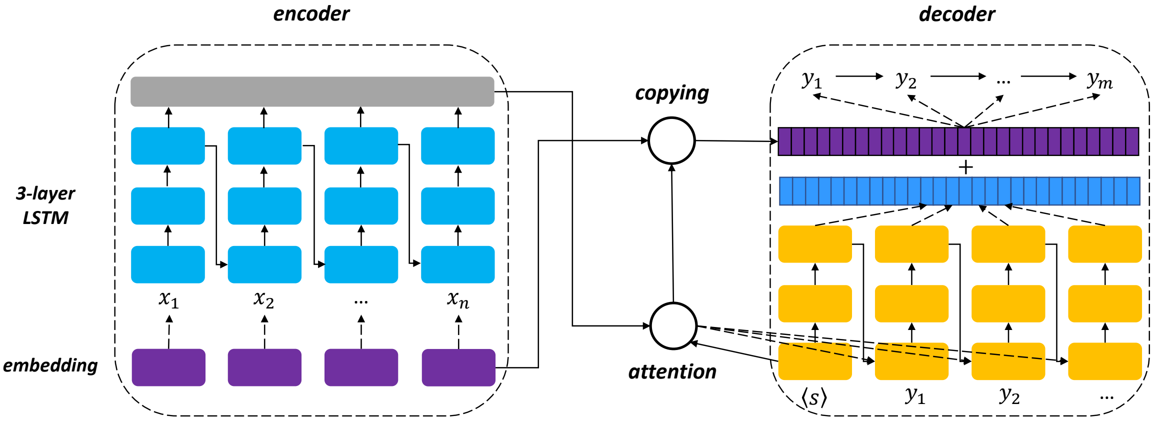

Figure 1: The encoder-decoder model architecture for the neural Open IE system

learning ⟨/arg2⟩". We obtain the input and output sequence pairs from highly confident tuples bootstrapped from a state-of-the-art Open IE system. Experiment results on a large benchmark dataset illustrate that the neural Open IE approach is significantly better than others in precision and recall, while also reducing the dependencies on other NLP tools.

The contributions of this paper are three-fold. First, the encoder-decoder framework learns the sequence-to-sequence task directly, bypassing other hand-crafted patterns and alleviating error propagation. Second, a large number of high-quality training examples can be bootstrapped from state-of-the-art Open IE systems, which is released for future research. Third, we conduct comprehensive experiments on a large benchmark dataset to compare different Open IE systems to show the neural approach's promising potential.

2 Methodology

2.1 Problem Definition

Let (X, Y) be a sentence and tuples pair, where $X = (x_1, x_2, ..., x_m)$ is the word sequence and $Y = (y_1, y_2, ..., y_n)$ is the tuple sequence extracted from X. The conditional probability of $P(Y|X)$ can be decomposed as:

$$\begin{aligned} P(Y|X) &= P(Y|x_1, x_2, ..., x_m) \\ &= \prod_{i=1}^{n} p(y_i|y_1, y_2, ..., y_{i-1}; x_1, x_2, ...x_m) \end{aligned} \quad (1)$$

In this work, we only consider the binary extractions from sentences, leaving n-ary extractions and nested extractions for future research. In addition, we ensure that both the argument and relation phrases are sub-spans of the input sequence. Therefore, the output vocabulary equals the input vocabulary plus the placeholder symbols.

2.2 Encoder-Decoder Model Architecture

The encoder-decoder framework takes a variable length input sequence to a compressed representation vector that is used by the decoder to generate the output sequence. In this work, both the encoder and decoder are implemented using Recurrent Neural Networks (RNN) and the model architecture is shown in Figure 1.

The encoder uses a 3-layer stacked Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997) network to covert the input sequence $X = (x_1, x_2, ...x_m)$ into a set of hidden representations $\mathbf{h} = (h_1, h_2, ..., h_m)$, where each hidden state is obtained iteratively as follows:

$$h_t = \mathbf{LSTM}(x_t, h_{t-1}) \quad (2)$$

The decoder also uses a 3-layer LSTM network to accept the encoder's output and generate a variable-length sequence Y as follows:

$$\begin{aligned} s_t &= \mathbf{LSTM}(y_{t-1}, s_{t-1}, c) \\ p(y_t) &= \mathbf{softmax}(y_{t-1}, s_t, c) \end{aligned} \quad (3)$$

where s_t is the hidden state of the decoder LSTM at time t, c is the context vector that is introduced later. We use the softmax layer to calculate the output probability of y_t and select the word with the largest probability.

An attention mechanism is vital for the encoder-decoder framework, especially for our neural

Open IE system. Both the arguments and relations are sub-spans that correspond to the input sequence. We leverage the attention method proposed by Bahdanau et al. to calculate the context vector c as follows:

$$\begin{aligned} c_i &= \sum_{j=1}^{n} \alpha_{ij} h_j \\ \alpha_{ij} &= \frac{\exp(e_{ij})}{\sum_{k=1}^{n} \exp(e_{ik})} \\ e_{ij} &= a(s_{i-1}, h_j) \end{aligned} \tag{4}$$

where a is an alignment model that scores how well the inputs around position j and the output at position i match, which is measured by the encoder hidden state h_j and the decoder hidden state s_{i-1}. The encoder and decoder are jointly optimized to maximize the log probability of the output sequence conditioned on the input sequence.

2.3 Copying Mechanism

Since most encoder-decoder methods maintain a fixed vocabulary of frequent words and convert a large number of long-tail words into a special symbol "⟨unk⟩", the copying mechanism (Gu et al., 2016; Gulcehre et al., 2016; See et al., 2017; Meng et al., 2017) is designed to copy words from the input sequence to the output sequence, thus enlarging the vocabulary and reducing the proportion of generated unknown words. For the neural Open IE task, the copying mechanism is more important because the output vocabulary is directly from the input vocabulary except for the placeholder symbols. We simplify the copying method in (See et al., 2017), the probability of generating the word y_t comes from two parts as follows:

$$p(y_t) = \begin{cases} p(y_t|y_1, y_2, ..., y_{t-1}; X) & \text{if } y_t \in V \\ \sum_{i:x_i=y_t} a_i^t & \text{otherwise} \end{cases} \tag{5}$$

where V is the target vocabulary. We combine the sequence-to-sequence generation and attention-based copying together to derive the final output.

3 Experiments

3.1 Data

For the training data, we used Wikipedia dump 20180101[3] and extracted all the sentences that are 40 words or less. OpenIE4 is used to analyze the sentences and extract all the tuples with binary relations. To further obtain high-quality tuples, we only kept the tuples whose confidence score is at least 0.9. Finally, there are a total of 36,247,584 ⟨sentence, tuple⟩ pairs extracted. The training data is released for public use at https://1drv.ms/u/s!ApPZx_TWwibImHl49ZBwxOU0ktHv.

For the test data, we used a large benchmark dataset (Stanovsky and Dagan, 2016) that contains 3,200 sentences with 10,359 extractions[4]. We compared with several state-of-the-art baselines including Ollie, ClausIE, Stanford OpenIE, PropS and OpenIE4. The evaluation metrics are precision and recall.

3.2 Parameter Settings

We implemented the neural Open IE model using OpenNMT (Klein et al., 2017), which is an open source encoder-decoder framework. We used 4 M60 GPUs for parallel training, which takes 3 days. The encoder is a 3-layer bidirectional LSTM and the decoder is another 3-layer LSTM. Our model has 256-dimensional hidden states and 256-dimensional word embeddings. A vocabulary of 50k words is used for both the source and target sides. We optimized the model with SGD and the initial learning rate is set to 1. We trained the model for 40 epochs and started learning rate decay from the 11^{th} epoch with a decay rate 0.7. The dropout rate is set to 0.3. We split the data into 20 partitions and used data sampling in OpenNMT to train the model. This reduces the length of the epochs for more frequent learning rate updates and validation perplexity computation.

3.3 Results

We used the script in (Stanovsky and Dagan, 2016)[5] to evaluate the precision and recall of different baseline systems as well as the neural Open IE system. The precision-recall curve is shown in Figure 2. It is observed that the neural Open IE system performs best among all tested systems. Furthermore, we also calculated the Area under Precision-Recall Curve (AUC) for each system. The neural Open IE system with top-5 outputs achieves the best AUC score 0.473, which is significantly better than other systems. Although the

[3] https://dumps.wikimedia.org/enwiki/20180101/

[4] https://github.com/gabrielStanovsky/oie-benchmark

[5] The absolute scores are different from the original paper because the authors changed the matching function in their GitHub Repo, but did not change the relative performance.

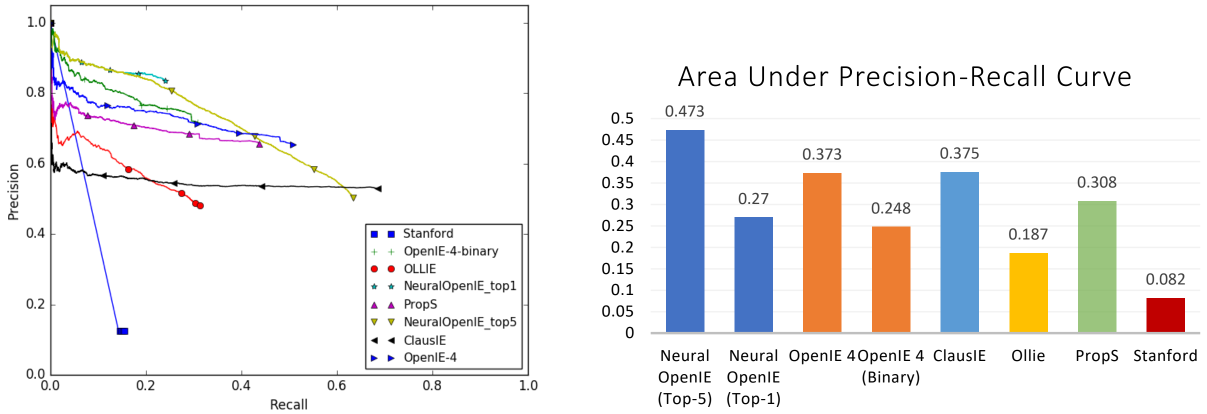

Figure 2: The Precision-Recall (P-R) curve and Area under P-R curve (AUC) of Open IE systems

neural Open IE is learned from the bootstrapped outputs of OPENIE4's extractions, only 11.4% of the extractions from neural Open IE agree with the OPENIE4's extractions, while the AUC score is even better than OPENIE4's result. We believe this is because the neural approach learns arguments and relations across a large number of highly confident training instances. This also indicates that the generalization capability of the neural approach is better than previous methods. We observed many cases in which the neural Open IE is able to correctly identify the boundary of arguments but OpenIE4 cannot, for instance:

Input	Instead , much of numerical analysis is concerned with obtaining approximate solutions while maintaining reasonable bounds on errors .
Gold	much of numerical analysis ||| concerned ||| with obtaining approximate solutions **while maintaining reasonable bounds on errors**
OpenIE4	much of numerical analysis ||| is concerned with ||| obtaining approximate solutions
Neural Open IE	much of numerical analysis ||| is concerned ||| with obtaining approximate solutions **while maintaining reasonable bounds on errors**

This case illustrates that the neural approach reduces the limitation of hand-crafted patterns from other NLP tools. Therefore, it reduces the error propagation effect and performs better than other systems especially for long sentences.

We also investigated the computational cost of different systems. For the baseline systems, we obtained the Open IE extractions using a Xeon 2.4 GHz CPU. For the neural Open IE, we evaluated performance based on an M60 GPU. The running time was calculated by extracting Open IE tuples from the test dataset that contains a total of 3,200 sentences. The results are shown in Table 1. Among the aforementioned conventional systems, Ollie is the most efficient approach which takes around 160s to finish the extraction. By using GPU, the neural approach takes 172s to extract the tuples from the test data, which is comparable with conventional approaches. As the neural approach does not depend on other NLP tools, we can further optimize the computational cost in future research efforts.

System	Device	Time
Stanford	CPU	234s
Ollie	CPU	160s
ClausIE	CPU	960s
PropS	CPU	432s
OpenIE4	CPU	181s
Neural Open IE	GPU	172s

Table 1: Running time of different systems

4 Related Work

The development of Open IE systems has witnessed rapid growth during the past decade (Mausam, 2016). The Open IE system was introduced by TEXTRUNNER (Banko et al., 2007) as the first generation. It casts the argument and relation extraction task as a sequential labeling problem. The system is highly scalable and extracts facts from large scale web content. REVERB (Fader et al., 2011) improved over TEXTRUNNER with syntactic and lexical constraints on binary relations expressed by verbs, which more than doubles the area under the precision-recall curve. Following these efforts, the second generation known as R2A2 (Etzioni et al., 2011) was developed based on REVERB and an argument identifier, ARGLEARNER, to better extract the arguments for the relation phrases. The first and second generation Open IE systems extract only relations that are mediated by verbs and ignore contexts. To alleviate these limitations, the third generation OLLIE (Mausam et al., 2012) was developed, which achieves better performance by extracting relations mediated by nouns, adjectives, and more. In addition, contextual information is also leveraged to improve the precision of extractions. All the three generations only consider binary extractions from the text, while binary extractions are not always enough for their semantics representations. Therefore, SRLIE (Christensen et al., 2010) was developed to include an attribute context with a tuple when it is available. OPENIE4 was built on SRLIE with a rule-based extraction system RELNOUN (Pal and Mausam, 2016) for extracting noun-mediated relations. Recently, OPENIE5 improved upon extractions from numerical sentences (Saha et al., 2017) and broke conjunctions in arguments to generate multiple extractions. During this period, there were also some other Open IE systems emerged and successfully applied in different scenarios, such as ClausIE (Del Corro and Gemulla, 2013) Stanford OPENIE (Angeli et al., 2015), PropS (Stanovsky et al., 2016), and more.

The encoder-decoder framework was introduced by Cho et al. and Sutskever et al., where a multi-layered LSTM/GRU is used to map the input sequence to a vector of a fixed dimensionality, and then another deep LSTM/GRU to decode the target sequence from the vector. Bahdanau et al. and Luong et al. further improved the encoder-decoder framework by integrating an attention mechanism so that the model can automatically find parts of a source sentence that are relevant to predicting a target word. To improve the parallelization of model training, convolutional sequence-to-sequence (ConvS2S) framework (Gehring et al., 2016, 2017) was proposed to fully parallelize the training since the number of non-linearities is fixed and independent of the input length. Recently, the transformer framework (Vaswani et al., 2017) further improved over the vanilla S2S model and ConvS2S in both accuracy and training time.

In this paper, we use the LSTM-based S2S approach to obtain binary extractions for the Open IE task. To the best of our knowledge, this is the first time that the Open IE task is addressed using an end-to-end neural approach, bypassing the hand-crafted patterns and alleviating error propagation.

5 Conclusion and Future Work

We proposed a neural Open IE approach using an encoder-decoder framework. The neural Open IE model is trained with highly confident binary extractions bootstrapped from a state-of-the-art Open IE system, therefore it can generate high-quality tuples without any hand-crafted patterns from other NLP tools. Experiments show that our approach achieves very promising results on a large benchmark dataset.

For future research, we will further investigate how to generate more complex tuples such as n-ary extractions and nested extractions with the neural approach. Moreover, other frameworks such as convolutional sequence-to-sequence and transformer models could apply to achieve better performance.

Acknowledgments

We are grateful to the anonymous reviewers for their insightful comments and suggestions.

References

Gabor Angeli, Melvin Jose Johnson Premkumar, and Christopher D. Manning. 2015. Leveraging linguistic structure for open domain information extraction. 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)*. Association for Computational Linguistics, Beijing, China, pages 344–354. http://www.aclweb.org/anthology/P15-1034.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *CoRR* abs/1409.0473. http://arxiv.org/abs/1409.0473.

Michele Banko, Michael J. Cafarella, Stephen Soderland, Matt Broadhead, and Oren Etzioni. 2007. Open information extraction from the web. In *Proceedings of the 20th International Joint Conference on Artifical Intelligence*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, IJCAI'07, pages 2670–2676. http://dl.acm.org/citation.cfm?id=1625275.1625705.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Doha, Qatar, pages 1724–1734. http://www.aclweb.org/anthology/D14-1179.

Janara Christensen, Mausam, Stephen Soderland, and Oren Etzioni. 2010. Semantic role labeling for open information extraction. In *Proceedings of the NAACL HLT 2010 First International Workshop on Formalisms and Methodology for Learning by Reading*. Association for Computational Linguistics, Los Angeles, California, pages 52–60. http://www.aclweb.org/anthology/W10-0907.

Luciano Del Corro and Rainer Gemulla. 2013. Clausie: Clause-based open information extraction. In *Proceedings of the 22Nd International Conference on World Wide Web*. ACM, New York, NY, USA, WWW '13, pages 355–366. https://doi.org/10.1145/2488388.2488420.

Oren Etzioni, Anthony Fader, Janara Christensen, Stephen Soderland, and Mausam Mausam. 2011. Open information extraction: The second generation. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence - Volume Volume One*. AAAI Press, IJCAI'11, pages 3–10. https://doi.org/10.5591/978-1-57735-516-8/IJCAI11-012.

Anthony Fader, Stephen Soderland, and Oren Etzioni. 2011. Identifying relations for open information extraction. In *Proceedings of the Conference of Empirical Methods in Natural Language Processing (EMNLP '11)*. Edinburgh, Scotland, UK.

Kiril Gashteovski, Rainer Gemulla, and Luciano Del Corro. 2017. Minie: Minimizing facts in open information extraction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Copenhagen, Denmark, pages 2630–2640. https://www.aclweb.org/anthology/D17-1278.

Jonas Gehring, Michael Auli, David Grangier, and Yann N Dauphin. 2016. A Convolutional Encoder Model for Neural Machine Translation. *ArXiv e-prints* .

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. 2017. Convolutional Sequence to Sequence Learning. *ArXiv e-prints* .

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor O.K. Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Berlin, Germany, pages 1631–1640. http://www.aclweb.org/anthology/P16-1154.

Caglar Gulcehre, Sungjin Ahn, Ramesh Nallapati, Bowen Zhou, and Yoshua Bengio. 2016. Pointing the unknown words. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Berlin, Germany, pages 140–149. http://www.aclweb.org/anthology/P16-1014.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Comput.* 9(8):1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M. Rush. 2017. Opennmt: Open-source toolkit for neural machine translation. *CoRR* abs/1701.02810. http://arxiv.org/abs/1701.02810.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Lisbon, Portugal, pages 1412–1421. http://aclweb.org/anthology/D15-1166.

Mausam. 2016. Open information extraction systems and downstream applications. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*. AAAI Press, IJCAI'16, pages 4074–4077. http://dl.acm.org/citation.cfm?id=3061053.3061220.

Mausam, Michael Schmitz, Robert Bart, Stephen Soderland, and Oren Etzioni. 2012. Open language learning for information extraction. In *Proceedings of Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CONLL)*.

Rui Meng, Sanqiang Zhao, Shuguang Han, Daqing He, Peter Brusilovsky, and Yu Chi. 2017. Deep keyphrase generation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Vancouver, Canada, pages 582–592. http://aclweb.org/anthology/P17-1054.

Ramesh Nallapati, Bowen Zhou, Cícero Nogueira dos Santos, aglar Gülehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. In *CoNLL*.

Harinder Pal and Mausam. 2016. Demonyms and compound relational nouns in nominal open ie. In *Proceedings of the 5th Workshop on Automated Knowledge Base Construction*. Association for Computational Linguistics, San Diego, CA, pages 35–39. http://www.aclweb.org/anthology/W16-1307.

Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Lisbon, Portugal, pages 379–389. http://aclweb.org/anthology/D15-1044.

Swarnadeep Saha, Harinder Pal, and Mausam. 2017. Bootstrapping for numerical open ie. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Association for Computational Linguistics, Vancouver, Canada, pages 317–323. http://aclweb.org/anthology/P17-2050.

Rudolf Schneider, Tom Oberhauser, Tobias Klatt, Felix A. Gers, and Alexander Löser. 2017. Analysing errors of open information extraction systems. In *Proceedings of the First Workshop on Building Linguistically Generalizable NLP Systems*. Association for Computational Linguistics, Copenhagen, Denmark, pages 11–18. http://www.aclweb.org/anthology/W17-5402.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Vancouver, Canada, pages 1073–1083. http://aclweb.org/anthology/P17-1099.

Gabriel Stanovsky and Ido Dagan. 2016. Creating a large benchmark for open information extraction. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Austin, Texas, pages 2300–2305. https://aclweb.org/anthology/D16-1252.

Gabriel Stanovsky, Jessica Ficler, Ido Dagan, and Yoav Goldberg. 2016. Getting more out of syntax with props. *CoRR* abs/1603.01648. http://arxiv.org/abs/1603.01648.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. *CoRR* abs/1409.3215. http://arxiv.org/abs/1409.3215.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, Curran Associates, Inc., pages 5998–6008. http://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf.

Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2014. Show and tell: A neural image caption generator. *CoRR* abs/1411.4555. http://arxiv.org/abs/1411.4555.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR* abs/1609.08144. http://arxiv.org/abs/1609.08144.