

Seq2RDF: An end-to-end application for deriving Triples from Natural Language Text

Yue Liu, Tongtao Zhang, Zhicheng Liang, Heng Ji, Deborah L. McGuinness

Department of Computer Science, Rensselaer Polytechnic Institute

Abstract. We present an end-to-end approach that takes unstructured textual input and generates structured output compliant with a given vocabulary. We treat the triples within a given knowledge graph as an independent graph language and propose an encoder-decoder framework with an attention mechanism that leverages knowledge graph embeddings. Our model learns the mapping from natural language text to triple representation in the form of **subject-predicate-object** using the selected knowledge graph vocabulary. Experiments on three different data sets show that we achieve competitive F1-Measures over the baselines using our simple yet effective approach. A demo video is included.

1 Introduction

Converting free text into usable structured knowledge for downstream applications usually requires expert human curators, or relies on the ability of machines to accurately parse natural language based on the meanings in the knowledge graph (KG) vocabulary. Despite many advances in text extraction and semantic technologies, there is yet to be a simple system that generates RDF triples from free text given a chosen KG vocabulary in *just one step*, which we consider an end-to-end system. We aim to automate the process of translating a natural language sentence into a structured triple representation defined in the form of **subject-predicate-object**, **s-p-o** for short, and build an end-to-end model based on an encoder-decoder architecture that learns the semantic parsing process from text to triple without tedious feature engineering and intermediate steps. We evaluate our approach on three different datasets and achieve competitive F1-measures outperforming our proposed baselines, respectively. The system, data set and demo are publicly available^{1,2}.

2 Our Approach

Inspired by the sequence-to-sequence model[5] in recent Neural Machine Translation, we attempt to use this model to bridge the gap between natural language and triple representation. We consider a natural language sentence $X = [x_1, \dots, x_{|X|}]$ as a source sequence, and we aim to map X to an RDF triple $Y = [y_1, y_2, y_3]$ with regard to **s-p-o** as a target sequence that is aligned with

¹ <https://github.com/YueLiu/NeuralTripleTranslation>

² <https://youtu.be/ssiQEDF-HHE>

a given KG vocabulary set or schema. Given DBpedia for example, we take a large amount of existing triples from DBpedia as ground truth facts for training. Our model learns how to form a compliant triple with appropriate terms in the existing vocabulary. Furthermore, the architecture of the decoder enables the model to capture the differences, dependencies and constraints when selecting s-p-o respectively, which makes the model a natural fit for this learning task.

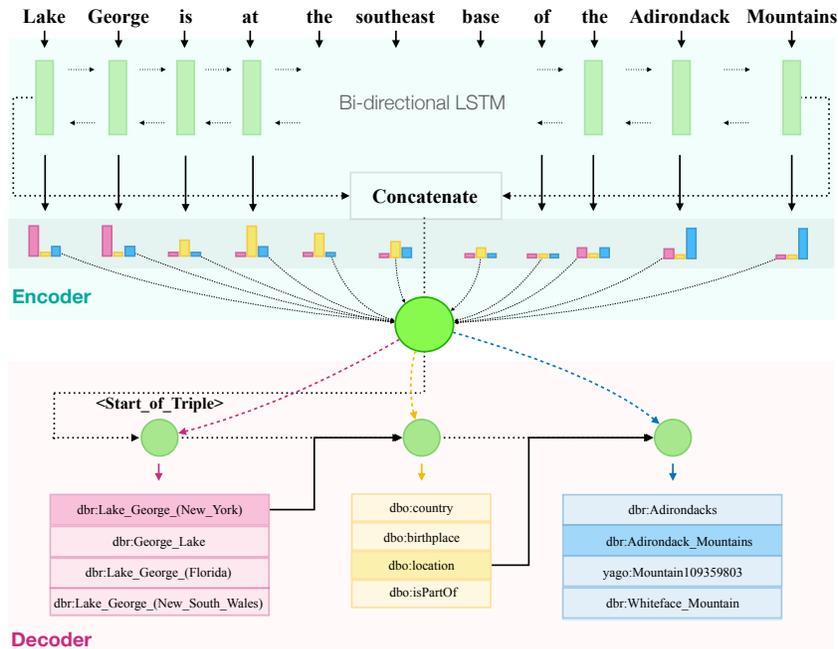


Fig. 1: Model Overview. Three colors (red, yellow, blue) represent the active attention during s-p-o decoding respectively. We currently only generate a single triple per sentence, leaving the generation of multiple triples per sentence for future work.

As shown in Figure 1, the model consists of an encoder taking in a natural language sentence as sequence input and a decoder generating the target RDF triple. The model pursues the maximized conditional probability

$$p(Y|X) = \prod_{t_d=1}^3 p(y|y_{<t_d}, X), \quad (1)$$

Both encoder and decoder are recurrent neural networks³ with Long Short Term Memory (LSTM) cells. We apply the attention mechanism that forces the model to learn to focus on specific parts of the input sequence when decoding, instead

³ We use `tf.contrib.seq2seq.sequence_loss` which is a weighted cross-entropy loss for a sequence of logits. We concatenate the last hidden output of forward and backward LSTM networks, the concatenated vector comes with fixed dimensions

of relying only on the last hidden state of the encoder. Furthermore, in order to capture the semantics of the entities and relations within our training data, we apply domain specific resources[2] to obtain the word embeddings and the TransE model[1] to obtain KG embeddings for entities and relations in the KG. We use these pre-trained Word embeddings and KG embeddings for entities and relations to initialize the encoder and decoder embedding matrix, respectively, and results show that this approach improves the overall performance.

3 Experiments

Data Sets We ran experiments on two public datasets NYT⁴ [4], ADE⁵ with selected vocabularies and a Wiki-DBpedia dataset that is produced by distant supervision⁶. For data obtained by distant supervision, the test set is manually labeled to ensure its quality. Each data set is an annotated corpus with corresponding triples in the form of either **s-p-o** or **entity mentions** and **relation types** at the sentence level. Details are available on our GitHub page.

Text	Berlin is the capital city of Germany.
Triple	dbr:Germany dbo:capital dbr:Berlin

Table 1: Example annotated pair with distant supervision on Wiki-DBpedia

Evaluation Metrics We consider pipeline-based approaches that combine Entity Linking (EL) and Relation Classification (RC) as state of the art. We propose several baselines with combined outputs from state-of-the-art EL⁷ and RC for evaluation. We use F1-measure to evaluate triple generation (an output is considered correct only if **s-p-o** are all correct) in comparison with the baselines.

Baselines We implement multiple baselines including a classical supervised learning using simple Lexical features, a state-of-the-art recurrent neural network (RNN) approach with LSTM [3] and one with a Gate Recurrent Unit (GRU) variant. Then we evaluate the performance on triple generation with results combining EL and RC. The hyper-parameters in our model are tuned with 10-fold cross-validation on the training set according to the best F1-scores. We applied the same settings to the baselines. The details regarding the parameters and settings are available on our GitHub page for replication purposes.

4 Result Analysis

We achieve the best F1 Measure of 84.3 on the triple generation from Table 2. Note that the baseline approaches that we implemented are pipeline-based, and thus they are very likely to propagate errors to downstream components. However, our model merges the two different tasks of EL and RC into one during the

⁴ New York Times articles: <https://github.com/shanzhenren/CoType>

⁵ Adverse drug events: <https://sites.google.com/site/adecorpus>

⁶ http://deepdive.stanford.edu/distant_supervision

⁷ Stanford, Domain specific NER

decoding, which composes a major advantage over pipeline-based approaches that usually apply separate models on EL and RC. The most common errors are caused by **Out of vocabulary** and **Noise from overlapping relations** in text. As we do not cover all rare entity names or consider multiple triple situations, these errors are valid in some sense.

Tasks	NYT	ADE	Wiki-DBpedia
Metric	F1-Measure	F1-Measure	F1-Measure
EL+Lexical	36.8	61.4	37.8
EL+LSTM	58.7	70.3	65.5
EL+GRU	59.8	73.2	67.0
Seq2Seq	64.2	73.4	73.5
S+A+W+G	71.4	79.5	84.3

Table 2: Cross-dataset comparison on triple generations. Seq2Seq denotes the implementation of Seq2Seq without any attention mechanism and pre-trained embeddings; A denotes attention mechanism; W and G denote pre-trained word embeddings for the encoders and KG embeddings for the decoders, respectively.

5 Conclusions and Future Work

We present an end-end system for translating a natural language sentence to its triple representation. Our system performs competitively on three different datasets and our assumption on enhancing the model with pre-trained KG embeddings improves performance across the board. It is easy to replicate our work and use our system following the demonstration. In the future, we plan to re-design the decoder and enable the generation of multiple triples per sentence.

Acknowledgement This work was partially supported by the NIEHS Award 0255-0236-4609 / 1U2CES026555-01.

References

1. Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: Advances in neural information processing systems. pp. 2787–2795 (2013)
2. Liu, Y., Ge, T., Mathews, K., Ji, H., McGuinness, D.: Exploiting task-oriented resources to learn word embeddings for clinical abbreviation expansion. Proceedings of BioNLP 15 pp. 92–97 (2015)
3. Miwa, M., Bansal, M.: End-to-end relation extraction using lstms on sequences and tree structures. arXiv preprint arXiv:1601.00770 (2016)
4. Ren, X., Wu, Z., He, W., Qu, M., Voss, C.R., Ji, H., Abdelzaher, T.F., Han, J.: Cotype: Joint extraction of typed entities and relations with knowledge bases. In: Proceedings of the 26th International Conference on World Wide Web. pp. 1015–1024 (2017)
5. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. In: Advances in neural information processing systems. pp. 3104–3112 (2014)