End-to-end Representation Learning for Question Answering with Weak Supervision

Daniil Sorokin and Iryna Gurevych

Ubiquitous Knowledge Processing Lab (UKP-TUDA)
Department of Computer Science, Technische Universität Darmstadt
www.ukp.tu-darmstadt.de

Abstract. In this paper we present a factoid question answering system for participation in Task 4 of the QALD-7 shared task. Our system is an end-to-end neural architecture for learning a semantic representation of the input question. It iteratively generates representations and uses a convolutional neural network (CNN) model to score them at each step. We take the semantic representation with the highest final score and execute it against Wikidata to retrieve the answers. We show on the Task 4 data set that our system is able to successfully generalize to new data.

Keywords: Semantic web, Question-answering, Representation learning, Convolutional neural networks, Semantic parsing, Weak supervision

1 Introduction

QALD is a series of international competitions on mapping natural language questions to knowledge base queries [9]. The goal of the competitions is to provide a benchmark for natural language based interfaces to knowledge bases.

In this paper, we present a system for Task 4 of the QALD-7 shared task, "English question answering over Wikidata". The task is formulated as follows: given a natural language question, translate it into a structured query in SPARQL that can be executed against Wikidata to obtain the answer to the question. The provided training data set for Task 4 consists of 100 natural language questions that require as an answer either a set of Wikidata items, a number or a date. In our system, we implement a semantic parsing approach [3], that is, we produce semantic representations for natural language questions that are then deterministically converted into SPARQL queries and executed against Wikidata.

Multiple successful question answering systems were presented in the previous QALD competitions (see for example the overview in [9]). Many systems (e.g. [2, 3, 5]) have constructed semantic representations for input questions and relied on trained models with manually defined features to select the correct one. Recently, end-to-end neural network approaches were introduced that can process simple questions that only need a single semantic relation to be answered (e.g. [4]).

In our approach, we also rely on a neural network model to disambiguate the question. Our main contribution is an end-to-end iterative generation of multi-relational semantic representations that integrates a neural representation scorer based on CNNs. The end-to-end neural architecture doesn't need handcrafted features or heavy pre-processing that are required in other approaches. It automatically learns a correspondence between structural and lexical features of a semantic representation and a natural language question. Thus, our approach can better generalize to new unseen questions than controlled language systems or approaches based on manually defined features.

2 System architecture

2.1 Entity linking

We tokenize the input and add part-of-speech tags to it with the Stanford CoreNLP toolkit [8]. Afterwards, we extract token fragments using a set of regular expression rules that match all sequences of nouns with adjacent modifiers. We look up the extracted fragments in Wikidata by comparing them to labels of the Wikidata items. Following the approach in [1], we sort the retrieved list of items by the combination of the Levenshtein distance between the fragment and the item label and the integer part of the item ID. We select the top candidate for each fragment as the final linking. For example, in the question "What was the first Queen album?", we recognize entities "Queen" and "album".

2.2 Iterative representation generation

We develop a representation generation procedure that defines what kind of representations can be constructed. Our semantic representations (see Figure 1) consist of a question variable node (shaded circle), entities with a fixed Wikidata ID (rectangles), constraints (rounded rectangles) and Wikidata relations (labeled arrows). When we evaluate a semantic representation against Wikidata, we retrieve all entities that can take the place of the question variable node, so that all relations and constraints hold. That is, the question variable node denotes the answer to the question.

We iteratively generate candidate semantic representations of the question using a set of actions which can be applied at each step, starting with an empty representation that contains only a question variable. We define three types of

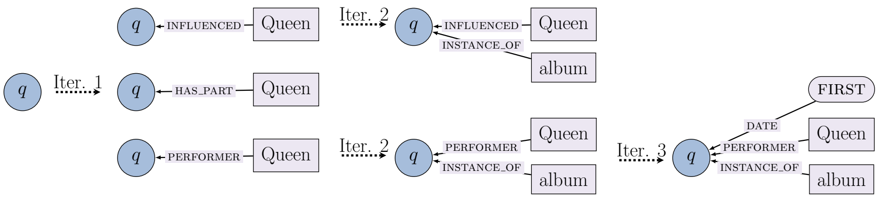

Fig. 1. Generating candidate representations for "What was the first Queen album?"

Action	Conditions	Action description
ADD_RELATION	LEN$(E) > 0$	Queries Wikidata for relations R that exist for $e, e \in E$, and creates a new representation for each $r, r \in R$
ADD_TEMP _CONSTRAINT	LEN(RELATIONS$(s)) > 0 \wedge$ LEN$(temp_markers \cap Q) > 0$	Creates a new representation with a constraint that the answer is the last or the first entity in a temporally sorted list
ADD_NUM _CONSTRAINT	LEN(RELATIONS$(s)) > 0 \wedge$ CONTAINS$(Q, number)$	Creates a new representation with an added relation that has a numeric argument

Table 1. The list of actions defined for the iterative representation generation process (E–list of entities, Q–list of question tokens, s–current semantic representation)

actions for representation generation: ADD_RELATION, ADD_TEMP_CONSTRAINT, ADD_NUM_CONSTRAINT. The defined actions constrain the space of possible semantic representations and make the search for the correct representation tractable. Each action adds the new representations to the list of candidates.

For each action, we define conditions that must be satisfied in order for the action to be applied at the current step (see Table 1). The conditions control the flow of the representation generation procedure. For example, at the first iteration in Figure 1 we apply the ADD_RELATION action, since it is the only action that can be performed on a empty representation. The result is one representation for each relation that exists for the entity "Queen" (Figure 1 shows only three). It is followed by another application of ADD_RELATION since there is a second entity in the question and finally ADD_TEMP_CONSTRAINT can be applied at the third iteration step because of a temporal marker "first" in the question. We check that each candidate representation is valid and representations that don't produce answers are not further expanded.

We perform a beam search to further reduce the space of considered representations for each question: after each iteration, we score the candidate representations with the neural scorer and select the top 10 for the next step. At the end, we select the representation with the highest score as the final output.

The neural scorer uses a CNN-based model to encode both the question and a candidate representation into a fixed-size semantic vector (see Figure 2). We compare the question vector with the vectors of the candidate representations using the cosine measure to produce scores. We choose CNNs as a basis for our model, since they have proven to be successful for question answering [2, 4].

To encode a question, we use character trigrams as suggested in [6]. Each token is represented as a binary vector where we set the positions that correspond to the trigrams in the token to 1. The token vectors are then processed by the CNN layer. The result of the CNN layer is processed by a max pooling layer

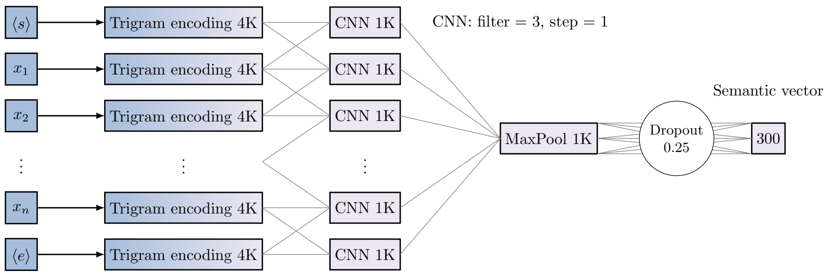

Fig. 2. The architecture of the CNN-based encoder (number represent dimensionality)

and then transformed by a fully connected layer with the tanh non-linearity to produce a semantic vector that encodes the question.

To encode a candidate representation, we first break it into individual relations. We tokenize the relation labels and use them as input to the same CNN-based encoder to produce a semantic vector for each relation.[1] To get a single vector for the whole representation, we apply another max pooling operation on the set of the relation vectors. The final semantic vector for a candidate representation encodes the most prominent features of the relations that it contains.

We use weak supervision in the form of question-answer pairs as suggested in [3] to train the neural network model. Weak supervision can provide more training data than available in the form of manually annotated semantic representations. We take the WebQuestions data set [3] which contains 3778 questions and manually retrieved answers. To get pairs of questions and semantic representations for model training, we run our representation generation procedure on each question. Instead of scoring the representations, we evaluate each representation against Wikidata and compare the extracted answers to the answers in the data set. The representations that result in at least a partially complete answer are stored as positive training instances. We use up to 20 incorrect representations to compute the loss (the Kullback-Leibler divergence) at each iteration during training. The neural network model is trained with the Adam optimizer [7].

3 Preliminary evaluation and Conclusions

In Table 2, we report preliminary evaluation results on the training data set for Task 4 of the QALD-7 Shared Task using the metrics from [9]. The "Upper bound" shows the results with an oracle neural scorer that always chooses the correct representation. Our model was not trained on this data set and, therefore, the reported results represent an expected generalization error of our system.

In this paper, we have presented an end-to-end system that produces semantic representations for natural language questions and evaluates them on Wikidata.

[1] All weights of the neural network model are shared in both cases, thus the vector encodings for questions and semantic representations are learned jointly.

	Processed	Right	Partially right	Avg. Precision	Avg. Recall	F1	Global F1
Our system	80	25	36	0.3507	0.4318	0.3640	0.2912
Upper bound	80	47	30	0.7602	0.8980	0.7266	0.5812

Table 2. Evaluation results on the QALD-7 Task 4 training (100 questions)

Our system produces Wikidata items as answers and can successfully process more than 50% of the questions in the QALD-7 Task 4 data set.

4 Acknowledgments

This work has been supported by the German Research Foundation as part of the QA-EduInf project (grant GU 798/18-1 and grant RI 803/12-1). We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Tesla K40 GPU used for this research.

References

1. Ahmad Aghaebrahimian and Filip Jurčíček. Open-domain Factoid Question Answering via Knowledge Graph Search. In *Proceedings of 2016 NAACL Human-Computer Question Answering Workshop*, pages 22–28, 2016.
2. Junwei Bao, Nan Duan, Zhao Yan, Ming Zhou, and Tiejun Zhao. Constraint-Based Question Answering with Knowledge Graph. In *Proceedings of COLING*, pages 2503–2514, 2016.
3. Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. Semantic Parsing on Freebase from Question-Answer Pairs. In *Proceedings of EMNLP*, pages 1533–1544, 2013.
4. Li Dong, Furu Wei, Ming Zhou, and Ke Xu. Question Answering over Freebase with Multi-Column Convolutional Neural Networks. In *Proceedings of ACL*, pages 260–269, 2015.
5. Sherzod Hakimov, Christina Unger, Sebastian Walter, and Philipp Cimiano. Applying Semantic Parsing to Question Answering over Linked Data: Addressing the Lexical Gap. In *Proceedings of NLDB*, pages 103–109, 2015.
6. Po-sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. Learning Deep Structured Semantic Models for Web Search using Clickthrough Data. In *Proceedings of CIKM*, pages 2333–2338, 2013.
7. Diederik Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. *arXiv preprint*, 2014.
8. Christopher D. Manning, John Bauer, Jenny Finkel, Steven J. Bethard, Mihai Surdeanu, and David McClosky. The Stanford CoreNLP Natural Language Processing Toolkit. In *Proceedings of ACL*, pages 55–60, 2014.
9. Christina Unger, Corina Forascu, Vanessa Lopez, Axel-Cyrille Ngonga Ngomo, Elena Cabrio, Philipp Cimiano, and Sebastian Walter. Question answering over linked data (QALD-5). In *CEUR Workshop Proceedings*, volume 1391, 2015.