Continuous Learning for Question Answering

Anselmo Peñas[1], Mathilde Veron[2], Camille Pradel[3], Arantxa Otegi[4], Guillermo Echegoyen[1], Alvaro Rodrigo[1]

[1] Universidad Nacional de Educación a Distancia (UNED), Juan del Rosal 16,
28040 Madrid, Spain
{anselmo, gblanco, alvarory}@lsi.uned.es
[2] LIMSI, CNRS, Université Paris-Saclay, Campus universitaire, bât. 507 rue du Belvédère
F-91405 Orsay Cedex, France
mathilde.veron@u-psud.fr
[3] Synapse Développement, 7 boulevard de la Gare
31500 Toulouse, France
camille.pradel@synapse-fr.com
[4] Ixa NLP Group, University of the Basque Country UPV/EHU, Manuel Lardizabal 1,
20018 San Sebastian, Spain
arantza.otegi@ehu.eus

Abstract. We consider the problem of answering natural language questions over a Knowledge Graph, in the case of systems that must evolve over time in a production environment. One of the key issues is that we can expect that the system will receive questions that cannot be answered with the current state of the Knowledge Graph. We discuss here the challenges we need to address in this scenario and the expected behavior of this kind of Lifelong learning system. We also suggest a first task to address this problem and a possible procedure to build a benchmark.

1 Scenario

We focus on the problem of Question Answering (QA) over Knowledge Bases: the task of building an interpretation to a Natural Language (NL) question, according to a Knowledge Base (KB), Database (DB) or Knowledge Graph (KG). Without much loss of generality, we will assume the KB has the form of a KG for the purposes of our study.

The general task is defined as follows: Given a NL question, produce a query in a formal language (e.g. SPARQL) to return from the KG the correct answers to the question.

In this setting, the aim of this work is to study the problem of answering natural language questions over a Knowledge Graph, in the case of systems that must evolve over time in a production environment. One of the key issues is that we can expect that the system will receive questions that cannot be answered with the current state of the Knowledge Graph.

The first challenge, then, is the detection of this situation: detect if the user's utterance can be mapped or not into an explicit knowledge subgraph. Maybe the answer is not in the KG because of a lack in its population (i.e. some resources / instances or property values are missing). But in the general case, it could happen that the KG doesn't contain information about classes and properties that might be of new interest for the user. In this case, a good system, which can evolve over time and adapt to these "surprise" new situations, should be able to look for resources outside, in order to extract and populate the KG, not only with new instances but also new classes and new properties.

This problem has already captured the attention of some researchers such as Mazumder and his colleagues [1] who presented the LiLi system, although in that work, the problem is only addressed partially and without a previous definition of the complete general scenario. In particular, queries to the LiLi system are just single triples, reducing to the trivial case the problem of deciding whether the answer to a question is in the KG or not. It simplifies also the problem of detecting the pieces of knowledge that have to be added to the KG. The option taken for LiLi system for enriching the KG is to ask the user for some missing pieces of knowledge and try to find strategies to infer some others.

However, in the general scenario of complex NL Question Answering over KGs these decisions are not trivial. If a system does not get an answer to a question, it could be due to several factors, including some error in the process of NL interpretation (e.g. Entity Linking).

We are agnostic with respect to the strategy and the role of users to address this problem. But this is a hard problem and we envisage that it should be solved in collaboration with users, especially for the explicit or implicit assessment of new information correctness, and the decision to incorporate it into the KG.

For the purposes of clarifying the scenario, consider the following example: Assume a cooking scenario where we have classes, such as recipe or ingredient, with instances such as mousse or milk, respectively, together with properties such as a has_ingredient relation between objects of recipe class and objects of ingredient class. Imagine now that the user asks something that cannot be answered with the current state of the KG: "How long do I need to put my mousse in the fridge?". We expect a system to follow a procedure similar to:

1. The system must determine that it cannot answer this question with the KG.
2. Then, the system needs to answer this question by making use of external resources (it can be done by collaborating with a user [4][5], after processing a large document collection [6], by querying the web [7], or through a combination of them). Imagine that finally the piece of text accepted says: "Chill the mousse in the refrigerator for 2 hours".
3. The system must identify that the KG has not an explicit relation between a recipe and a time duration similar to "time in the fridge". So, it decides to declare a new relation "fridge-duration" with range on recipes and domain on time duration.
4. The system adds a new triple for this relation: <mousse, fridge-duration, "P0DT2H"^^xsd:duration>
5. Once the system has confirmation of a relation that must be added to the KG, together with a good example, then, the system must go through all instances of the involved class (i.e. recipes) and use the external sources to try to populate this new relation in the KG. Depending on the approach, it could be

interesting for the system to maintain other information such as the natural language statements used to express the relation, etc.

This is a very ambitious scenario that poses many interesting research questions that can be addressed in many different ways, with many different strategies. The first effect is that current state of the art in QA over KGs cannot address this problem because most of the systems are built over the assumption that the KG contains the answer to the question [2][3].

Therefore, in our opinion, this is the first challenge we should address in the described scenario: the detection of situations where the KG does not contain the required pieces of information.

2 A proposal for a shared task

According to the scenario described above, we propose to address a first task as the first step towards the development of continuous learning QA systems. Since we are working with KGs, we can assume without loss of generalization, that we have objects, types or classes attached to the objects, literals, and relations or properties between objects and literals. Therefore, the missing pieces of information that impede the mapping of a user utterance into a KG subgraph can be any of them.

The task, then, is: Given a KG, and an external resource such as a text collection in the same domain:

1. decide whether a user utterance in Natural Language can be mapped or not into the KG and, if this is the case,
2. determine through which ones of the following ways the system must find a strategy to enrich the knowledge graph:
   a. Declare a new class (or object type). For example: utensil.
   b. Add a new instance of a class. For example: rolling pin.
   c. Declare a new relation. For example: needs_utensil.
   d. Add a new triple to the KG. For example: <shortbread, needs_utensil, rolling pin>.

The task is being expressed as a classification problem, which allows a wide range of different techniques. At the end, we would be training a set of five binary classifiers.

To test this task, we would need a benchmark dataset. There are several ways in which we can build such dataset, but perhaps the easiest one is to take advantage of current QA datasets with NL questions that must be answered according to a KG. For each question we can remove some elements form the KG that impede the system to answer the question. These elements can be from a particular instance up to a complete relation. Thus, somehow what we are expecting from the system is to guess the kind of information that is missing in the KG.

Acknowledgments. This work has been funded by the Spanish Research Agency (Agencia Estatal de Investigación), the “Agence Nationale pour la Recherche” (ANR, France) and the LIHLITH project (PCIN-2017-085 / AEI; PCIN-2017-118 / AEI;

ANR-17-CHR2-0001-03; ANR-17-CHR2-0001-04) inside the ERA-Net CHIST-ERA framework.

References

1. Sahisnu Mazumder, Nianzu Ma, Bing Liu. Towards a Continuous Knowledge Learning Engine for Chatbots. arXiv:1802.06024 [cs.CL], 16 Feb. 2018.
2. Konrad Hoffner, Sebastian Walter, Edgard Marx, Ricardo Usbeck, Jens Lehmann, and Axel-Cyrille Ngonga Ngomo. Survey on challenges of question answering in the semantic web. Semantic Web, 8(6):895–920, 2017.
3. Dennis Diefenbach, Vanessa Lopez, Kamal Singh, and Pierre Maret. Core techniques of question answering systems over knowledge bases: a survey. Knowledge and Information Systems, 55(3):529–569, 2018.
4. Jiwei Li, Alexander H. Miller, Sumit Chopra, Marc'Aurelio Ranzato, and Jason Weston. Learning through dialogue interactions by asking questions. arXiv preprint arXiv:1612.04936, 2016.
5. Jason E Weston. Dialog-based language learning. Advances in Neural Information Processing Systems, p. 829-837, 2016.
6. Sahisnu Mazumder and Bing Liu. Context-aware path ranking for knowledge base completion. IJCAI, 2017.
7. Robert West, Evgeniy Gabrilovich, Kevin Murphy, Shaohua Sun, Rahul Gupta, Dekang Lin. Knowledge base completion via search-based question answering. Proceedings of the 23rd international conference on World wide web, ACM, p. 515-526, 2014.