Procesamiento del Lenguaje Natural, Revista nº 63, marzo de 2019, pp. 121-128 recibido 29-03-2019 revisado 14-05-2019 aceptado 14-05-2019

Benchmarking Entity Linking for Question Answering over Knowledge Graphs

Evaluación del Enlazado de Entidades para Sistemas Pregunta-Respuesta sobre Grafos de Conocimiento

Guillermo Echegoyen, Álvaro Rodrigo, Anselmo Peñas
Natural Language Universidad Nacional de Educación a Distancia (UNED)
{gblanco, alvarory, anselmo}@lsi.uned.es

Abstract: Entity Linking (EL) is the process of anchoring a part of a question to a node (entity) already known in a Knowledge Base (KB). Although EL has been widely studied with large documents such as webpages, there have not been studies about its impact on Question Answering (QA). In this paper, we study benchmarks for QA and how they are composed, providing insights about its suitability for a real evaluation about the state of the art in QA, specillay if we want to take into account the subtask of EL. We propose a semi-automatic method to generate an EL dataset linked to the QA task taking advantage of pre-existing QA datasets. We apply this method to benchmarking QA collections, analyze the results and release the created dataset to the research community, including a subset focused on complex EL in QA. We believe that EL effectiveness in the context of QA can be better assessed through the use of the proposed dataset.
Keywords: Question Answering, Knowledge Bases, Entity Linking, DBPedia

Resumen: El Enlazado de Entidades (EE) consiste en asociar partes de un texto con nodos de una Base de Conocimiento (BC). A pesar de que se ha prestado bastante atención a la tarea de EE en documentos, apenas hay estudios relativos a su impacto en el campo de la Búsqueda de Respuestas (BR). En este trabajo estudiamos la composición de varias colecciones de BR y realizamos varias observaciones relativas a su adecuación para evaluar sistemas BR, especialmente en lo relativo a realizar EE. También proponemos un método semiautomático para crear colecciones de EE en el contexto de BR reaprovechando colecciones existentes de BR. Posteriormente, aplicamos nuestro método a varias colecciones actuales de BR, analizamos los resultados obtenidos y ponemos a disposición de la comunidad científica la colección de EE generada, incluyendo un subconjunto que contiene los ejemplos donde es más difícil realizar EE. Consideramos que la disponibilidad de esta nueva colección permitirá una mejor evaluación de la tarea de EE en el contexto de la BR.
Palabras clave: Búsqueda de Respuestas, Bases de Conocimiento, Enlazado de Entidades, DBPedia

1 Introduction

Digital information grows everyday. As a consequence, users must face an enormous amount of data when searching for specific information. A great amount of this information is stored as structured data inside Knowledge Bases (KB), databases, etc. A desirable feature of such structured data is the ability to query it in a user-friendly manner, or even better with natural language questions.

Question Answering (QA) systems over Knowledge Graphs (KG) receive Natural Language questions and return answers from structured databases such as DBpedia (Lehmann et al., 2014) or Freebase (Bollacker et al., 2008). For this purpose, these systems must translate the natural language question into a structured query understandable using, for instance SPARQL (Shekarpour et al., 2016).

One challenge of these QA systems is to link parts of the Natural Language question to their corresponding nodes in the Graph.

ISSN 1135-5948. DOI 10.26342/2019-63-13

When the element to be linked is an entity, the task is called Entity Linking (EL) and is usually at the upfront of any Question Answering System (Unger et al., 2014). More formally, EL is the process of mapping a given string containing an entity, onto a resource of the target KB.

In general, mapping an entity based on a string comprises two steps: (1) recognizing the entity mention in the text and (2) disambiguating it. Let us consider the following example[1]: given the question *Who all have been a manager of english under twenty one football team?*, we would like to map the string *english under twenty one football team* to its corresponding entity in a KG (we use DBPedia here), which in this case is *dbr:England_national_under-21_football_team*. To do so we need to recognize the entity mention (*english under twenty one*) and map it to the correct KB resource (disambiguation). In many cases, this process implies lemmatizating words (e.g. *English* → *England*), parsing numbers (*twenty one* → *21*), dealing with missing words (*national* is not present at all in the source question) or even considering alias with higher lexical gap.

The process of EL is just one part inside the QA pipeline, although very important: linking to a certain node will restrict the search process inside the KG, yielding poor results when the entity is linked incorrectly or simply not linked at all. Therefore, it is important to properly asses the impact of EL over QA Systems. For this purpose, we need a procedure for EL benchmarking in the context of QA. However, there is no collection, to the best of our knowledge, focused on evaluating EL for QA over KGs.

In this paper, we describe a novel method for building a collection oriented to evaluate EL in QA. Our method reuses collections of QA. So, we show the results of applying our method to common collections.

The main contributions of this work are:

- We characterize the main QA datasets
- We propose a semi-automatic method to generate an EL dataset linked to the QA task
- We create and release a large benchmark dataset for EL in QA
- We create and release a subset focused on complex EL in QA

We have found that the standard collections used to evaluate QA do not pay so much attention to the EL task. We show how the EL task over these collections is too easy given the nature of the collections. Thus, we consider that this work points out potential problems in the QA datasets. Moreover, we release a dataset focused on EL in QA, where the more complicated examples are already detected. This dataset can help other researchers to improve their systems.

The paper is structured as follows: Section 2 analyses the work done up to date with respect to Question Answering over Knowledge Graphs datasets. We thoroughly describe the task we propose the benchmark for. In section 3 the technical aspects of the dataset are explained, as well as its construction and peculiarities. Section 4 describes the proposed method to generate EL datasets and the baseline against which we measured it. Section 5 shows the results we obtained with our method and opens a debate about Entity Linking datasets. We further extend the discussion in section 6. In section 7 some details about the released resources are given. We finish the article with some conclusions and further research in section 8. A brief description of the scripts employed for every part of the process as well some pointers to the source code can be found on the appendix (A).

2 Related Work

We have not found collections of EL related to QA. This is why, in this Section, we focus on collections created for evaluating EL.

The Web People Search (WePS) Evaluations proposed a task similar to that of Word Sense Disambiguation (WSD), consisting in disambiguating person names based on internet results or wikipedia pages (Artiles et al., 2010). The organizers provided more than 4.5K documents with human annotations.

The Text Analisys Conference (TAC) organize an EL task since 2009. Systems participating in this task have to link entities in texts with entities in a KB. Organizers provide in each edition a new set of documents with human annotations (Mcnamee and T Dang, 2009).

These works try to link or disambiguate

[1] Taken from LC-QuAD (Trivedi et al., 2017), qid:1

entities given a big context (webpages or documents). However, in QA the available contextual information consists solely of a natural language question. This is why we now focus on benchkmarks related to EL with smaller contexts. This problem is known as short-text EL (Chen et al., 2018).

Webscope[2] is a good example of a collection for short-text EL. Webscope contains more than 2k search-engine queries where humans have identified links of entities to Wikipedia articles.

The dataset of the #Microposts2016 Named Entity rEcognition and Linking (NEEL) Challenge contains more than 6k tweets with entities annotated by humans and linked to DBpedia (Rizzo et al., 2016). In the scope of Twitter, there exists also the MSR-TEL dataset (Guo, Chang, and Kıcıman,). This dataset contains less than 1k tweets annotated with entities in the text linked to Wikipedia articles.

All these datasets are usefull for evaluating EL, but they cannot be applied to measure the impact of EL in the context of QA. Even though we could evaluate the EL task in a similar scenario to QA over KG, we still suffer from the lack of assessing the effect of EL performance in the final QA system. We think this fact hinders evaluating the impact of the EL task over the whole performance of a QA System and motivates our work in this paper.

3 *Description of the QA datasets*

There are many large datasets oriented to evaluate QA over KGs. In these datasetes, a sample or datapoint consists of a natural language question accompanied with the SPARQL Query that gathers the correct answers. The queries are attached to a particular KG such as DBPedia or Freebase. For instance, QALD (Unger et al., 2014) and LC-QuAD (Trivedi et al., 2017)) are two important datasets for QA over DBPedia. An example from QALD is depicted in listing 1.

Dataset creators employed different methods. In the case of QALD, the creators manually compiled the questions from real-world question and query logs (Unger et al., 2014). LC-QuAD, on the other hand, went for an automatic system that creates questions based on a few given templates and a target KG

[2]http://webscope.sandbox.yahoo.com/

```
{
  "id": "37",
  "query": {
    "sparql": "SELECT ?uri ?string WHERE{
         ?uri dbo:series dbr:The_Sopranos
         . ?uri dbo:seasonNumber 1 .
        OPTIONAL {?uri rdf-schema#label ?
        string . FILTER (lang(?string) =
        'en') } }"
  },
  "answers": {
    "answer": [{
      "uri": "http://dbpedia.org/resource
          /46_Long",
      "string": "46 Long"
    },{
      "uri": "http://dbpedia.org/resource
          /A_Hit_Is_a_Hit",
      "string": "A Hit Is a Hit"
    },
    ...
    {
      "uri": "http://dbpedia.org/resource
          /College_%28The_Sopranos%29",
      "string": "College (The Sopranos)"
    }]
  },
  "question": [
    {
      "string": "List all episodes of the
           first season of the HBO
          television series The Sopranos
          !",
      "language": "en"
    }
  ]
}
```

Listing 1: Simplified sample from the original QALD 1 dataset.

(Trivedi et al., 2017)

Table 1 shows some basic details, like the number of total questions, number of questions containing entities, average number of entities per question and total number of unique entities covered by each dataset. The difference in size between QALD 1 through 3 is notable, but the biggest one appears between LC-QuAD and the rest, being one order of magnitude. This difference accounts for the fact that LC-QuAD was semi automatically constructed. It is to be noted that the first edition of QALD features a rather low number of questions containing entities.

In Table 2 we depict the identified EL casuistry by dataset. The the first row of the Table shows the number of questions for which an exact match between the mention in the question and the resource is present. We find this number surprisingly high across all analysed datasets. Also, we acknowledge that the number of different EL cases

```
{
  "id": 2,
  "question_id": 37,
  "question": "List all episodes of the
      first season of the HBO television
      series The Sopranos!",
  "dbr": "http://dbpedia.org/resource/
      The_Sopranos",
  "mention": "The Sopranos"
}
```

Listing 2: Sample from QALD 1 dataset after processing for Entity Linking. The mention must exclude punctuation as it is not present in the resource.

in QALD grows with the editions of the task, being QALD-4 the most richer in terms of number of different cases. Given the size of LC-QuAD dataset, we have excluded it from the manual cases classification.

Based on the observation that lots of mentions can be directly inferred from the entity uri (DBPedia resource, or *dbr*), we propose a custom method to automatically extract the part of the question (mention) that gives the best hints to the database resource. We present this method in the following sections.

Dataset	Q	Q-E	Avg E	U. E
QALD-1	50	12	1.25	13
QALD-2	100	72	1.17	70
QALD-3	100	72	1.24	72
QALD-4	200	160	1.27	162
LC-QuAD	5000	5000	1.32	3962

Table 1: Number (Q)uestions, (Q-E) Questions with entities, (Avg E) Average entities per question and number of (U. E) Unique Entities by dataset.

4 Method for EL dataset generation

We propose a method for the generation of a new dataset for EL where a sample comprises: 1) the original natural language question; 2) one of the entities pertaining to the question and 3) the part of the questions that relates to the entity (mention). A sample is depicted in listing 2.

Since a question may contain various entities in the original QA datasets, our method can create several samples for the same question. To build each sample we need to:

1. Identify the mention of the entity in the question.
2. Disambiguate this mention according to the KG.

However, this last step is trivial because such information is already provided by the QA dataset itself. Therefore, we only need to focus on the task of mention detection. In this case, we can take the expected uri as a clue.

4.1 Trigram-based mention detection

We propose a method to construct and semi-automatically annotate an EL dataset based on any QA dataset that provides a set of natural language questions with their corresponding SPARQL queries that retrieves answers from a KG. The fundamental challenge we address here is *identifying the mention* in the Natural Language Question (NLQ) that tallies with a resource in the Knowledge Base. These mentions can be used later to aid in the disambiguation process of any EL system.

We are going to apply a strategy consisting in grouping contiguous highly overlapping trigrams character sets from the question, similar to labeling the mentions with a BIO scheme[3]. This way, we obtain a set of candidate mentions from which we choose the most promising one and apply a simple alignment to clean it.

Also, as a preprocessing step for every text, we apply a series of regular expressions that trim characters like commas and periods, substitute underscores and hyphens with spaces and trim multiple consecutive spaces (*clean* procedure used in alg. 1).

The proposed method is depicted by algorithm 1 that works as follows:

1. Apply the previously explained simple clean procedure both to the question and the entity.
2. Divide every token in trigrams and get the possible mentions. If no mention is found, return the word with the maximum number of matching trigrams (*find_mentions* procedure, alg. 2). The probability of a match between trigrams

[3]The BIO scheme suggests to learn classifiers that identify the **B**eginning, the **I**nside and the **O**utside of the text segments

EL Casuistry	QALD-1	QALD-2	QALD-3	QALD-4
Mentions identical to DBP uri	11	72	75	160
Typographic		1	1	1
Missing tokens in the mention		4	5	20
Additional tokens in the mention	3	1	1	1
Synonyms			1	1
Acronyms				2
Lexical variation	1	5	5	17
Lexical variation + reordering		1	1	
Total	15	84	89	203

Table 2: Identified Entity Linking Typographic quantities by dataset.

Algorithm 1: Dataset construction

```
Data: QuAD.json
Result: QuAD-EL.json
mentions ← []
for point in Questions do
    question ← clean(point.question)
    entities ← clean(point.entities)
    for entity in entities do
        best_score ← ∞
        best_candidate ← null
        candidates ←
         find_mentions(question,entity)
        for candidate in candidates do
            score ← distance(entity,
             candidate)
            if score < best_score or
             (score == best_score and
             len(candidate) <
             len(best_candidate))
             then
                best_score ← score
                best_candidate ←
                 candidate
            end
        end
        mentions ← mentions ∪
         best_candidate
    end
end
write(mentions)
```

sets is measured as the number of trigrams that match divided by the total number of trigrams.

3. Get the mention that minimizes Levenshtein distance (Levenshtein, 1966) to the entity with the minimum length (*distance* procedure).

Algorithm 2: Find Mentions

```
Data: question, entity
Result: mentions
mentions ← []
mention ← []
e_grams ← ngrams(entity)
best_prob ← 0
best_index ← null
best_score ← ∞
for word, index in question do
    w_grams ← ngrams(word)
    /* get matching grams      */
    max_common ← max(1,
     len(w_grams))
    common_tris ← 0
    for gram in w_grams do
        if gram in e_grams then
            common_tris ←
             common_tris +1
        end
    end
    prob ← common_tris/max_common
     if prob > 0.7 then
        mention ← mention ∪ word
    else
        if prob > best_prob then
            best_prob ← prob
            best_index ← index
        end
        mentions ← mentions ∪
         mention mention ← []
    end
    if mention is not Empty then
        mentions ← mentions ∪
         mention
    end
    if mentions is Empty then
        mentions ← question
         [best_index ]
    end
end
return mentions
```

4.2 Baseline mention detection

In order to asses the effectiveness of our method, we have also developed a simple and fast baseline method.

The method searches in the question for a contiguous span with size in tokens equal to that of the database resource such that the Levenshtein distance between the span and the resource is minimized.

This is a simple baseline that, even though it works only at the lexical level, is robust enough.

5 Results

We manually annotated and classified the cases by type for all the questions. Then, we evaluated the proposed method and the baseline. We acknowledge that the number of exact matches found is high on all datasets. This fact arises some questions about the quality of the datasets that will be discussed in the discussion section (sect. 6).

The obtained results are depicted on table 3. Because both methods yielded the same exact mention matches, that row is omitted from this table (and showed only in table 2). Table 3 shows the number of correctly extracted mentions by each method on each dataset, classified by EL case type. At a first sight, the proposed method out stands in all scenarios. From the second error type (*missing tokens in the mention*), we observe that the baseline does not adjust correctly when the question lacks part of the resource. It is clear that this happens because the method is only allowed to choose segments of the same size as the resource, yielding more words when it should not.

Among the failed questions we have found some questions from which a mention linking the entity and the natural language query cannot be established directly. In the process of constructing the LC-QuAD dataset some questions got transformed in a way that the SPARQL Query is unrelated to the question itself, although this accounts for a small part of the dataset (about 6 questions).

Regarding the other errors, the first and most obvious is related to spelling. In English there are no accents, in the automatic construction step most of them got stripped off, resulting in misspelled words. Also, there are lots of abbreviations and acronyms which hampers joining on matching trigrams without an expansion mechanism. When speaking or writing in many languages we usually inflect words, but on the other hand, DBPedia stores entities in its lemmatized form, rising the necessity to analyse the words further.

The *lexical variation* error type is a good example of this effect. Let us consider *Which monarchs of the United Kingdom were married to a German?*[4] as the source question, one necessary resource to answer the question is the one that relates to *German*, but the database resource is *dbr:Germany*. This implies that any given system must be able to select it as the mention.

For further details, refer to each annotated dataset, which contains the list of questions classified by error type.

6 Discussion about Entity Linking in QA Datasets

In our study, we have found that the number of questions containing the exact mention to the resource is significantly high (more than a 70% in each QALD dataset and an 80% in LC-QuAD). Thus, it is unclear if these datasets might be helpful for training QA systems working in real environments, where users may include questions using only a surname, part of the entity, etc.

We think that this issue arises as a consequence of how collections are created. All the analysed datasets automate parts of the process to some extent. While QALD datasets use logs and already available queries as the basement to formulate questions, LC-QuAD uses a slot-filling + templates technique to generate queries and then translate to NLQ.

We have also reviewed WebQuestions (Berant et al., 2013), which was created from non-experts using questions that begin with a wh-word and contain exactly one entity. The answers were searched using Freebase. In this dataset, 73% of questions contain the exact mention of the entity in Freebase. Therefore, these datasets do not pay too much attention to the problem of Entity Linking.

Given the big amount of cases where we can extract exact mention matches, we envision that a resource containing all the questions containing different cases could be valuable for the research community. It can be useful to assess the effectiveness of EL systems against a complex dataset. This will be

[4] Taken from QALD 3, qid: 38

Cases	QALD-1		QALD-2		QALD-3		QALD-4	
	B	T	B	T	B	T	B	T
Typographic			1/1	1/1	0/1	0/1	0/1	0/1
Missing tokens in the mention			0/4	4/4	1/5	4/5	1/20	14/20
Additional tokens in the mention	0/3	0/3	1/1	1/1	1/1	1/1	1/1	0/1
Synonyms					1/1	0/1	1/1	0/1
Acronyms							0/2	0/2
Lexical variation	0/1	0/1	5/5	5/5	5/5	5/5	16/17	15/17
Lexical variation + reordering			1/1	1/1	0/1	1/1		
Total	0/4	0/4	8/12	12/12	8/14	11/14	19/42	29/42

Table 3: Correct results obtained on (B)aseline and (T)rigrams based methods over each dataset (identical mentions ommited)

addressed in the next section.

Besides, all these collections usually contain KB resources that can be easily disambiguated, but in a real world scenario we can expect questions where there is not enough context to disambiguate the entity. A good example of this is when the name of the person that the question refers to cannot be disambiguated, i.e.: *What was the wealth of Aristoteles?*, *When did Armstrong die?*. This leads to a scenario where a QA system might ask for additional information about the mention before returning a possible resource. However, current datasets are not useful for such scenarios.

7 Release of datasets for EL in QA

One of the contributions we provide is the Entity Linking prepared version of each dataset, namely *LC-QuAD-EL* and *QALD-X-EL*. In the case of QALD based datasets from 1 to 4 we manually classified all the questions by error type, as well as annotate the correct mention. For LC-QuAD, given its size, we only annotated the mentions.

To construct each *EL* dataset, we have applied our method to automatically annotate all exact matches and manually correct the rest.

The contributed datasets follows two objectives: (1) Evaluate the proposed method itself and (2) Produce high quality and reliable datasets for evaluating this task. We have produced and openly released the following resources:

1. QALD datasets 1 through 4: We obtained some few hundred samples in total. The resources are named: QALD-X-EL, with X from 1 to 4.
2. LC-QuAD, that features 5K natural language questions accompanied with its corresponding SPARQL queries, was expanded to 6.6K samples, comprising 3.9K different resources. The final dataset is LC-QuAD-EL
3. Complex-EL4QA: A curated compilation of all the questions across all the analysed datasets that were not exact matches.

Table 4 shows the number of unique questions, unique entities covered as well as the total number of samples obtained for each Entity Linking dataset.

Dataset	U. Q.	U. E.	Total
QALD-1-EL	3	3	4
QALD-2-EL	11	11	12
QALD-3-EL	13	13	14
QALD-4-EL	38	40	45
LC-QuAD-EL	1204	997	1292
C-EL4QA	1307	1083	1469

Table 4: Number (U)nique questions, (U)nique entities covered and number of (T)otal.

8 Conclusions and Future Work

Entity Linking is still an open challenge in the NLP community and an important task in Question Answering. However, there is no specific collection for evaluating Entity Linking in QA.

In this paper, we introduce a benchmark to measure linkage accuracy between KB entities and mentions in a natural language question. In doing so, we classify common cases that add complexity to the task. Also, a

simple yet effective baseline approach is proposed and it can be used as the ground foundation of new, more complex systems. With this new benchmark we expect to help narrowing down the problem of EL intermediate step when answering questions.

Moreover, we deliver to the research community a new dataset that comprises all non-exact matched questions across all datasets that can be used to assess the effectiveness of the previous methods proposed over LC-QuAD, QALD and WebQuestions associated tasks. We also suggest to pay more attention to Entity Linking in QA collections.

We think the real world scenario where we can expect ambiguous questions deserves more attention. Also, we introduce the research question about how to evaluate the interaction with the user to capture the disambiguated version of the information needed.

Acknowledgments

This work has been partially funded by the Spanish Research Agency (Agencia Estatal de Investigación) LIHLITH project (PCIN-2017-085/AEI) in the framework of EU ERA-Net CHIST-ERA and RTI2018-096846-B-C21 (MCIU/AEI/FEDER,UE).

References

Artiles, J., A. Borthwick, J. Gonzalo, S. Sekine, and E. Amigó. 2010. Weps-3 evaluation campaign: Overview of the web people search clustering and attribute extraction tasks. In *CLEF 2010*.

Berant, J., A. Chou, R. Frostig, and P. Liang. 2013. Semantic parsing on freebase from question-answer pairs. In *EMNLP 2013*, pages 1533–1544.

Bollacker, K., C. Evans, P. Paritosh, T. Sturge, and J. Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250. AcM.

Chen, L., J. Liang, C. Xie, and Y. Xiao. 2018. Short text entity linking with fine-grained topics. In *CIKM '18*, pages 457–466, New York, NY, USA. ACM.

Guo, S., M.-W. Chang, and E. Kıcıman. To link or not to link? a study on end-to-end tweet entity linking. In *Proceedings of NAACL-HLT*, pages 1020–1030.

Lehmann, J., R. Isele, M. Jakob, A. Jentzsch, D. Kontokostas, P. Mendes, S. Hellmann, M. Morsey, P. Van Kleef, S. Auer, and C. Bizer. 2014. DBpedia - A Large-scale, Multilingual Knowledge Base Extracted from Wikipedia. *Semantic Web Journal*, 6.

Levenshtein, V. 1966. Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Soviet Physics Doklady*, 10.

Mcnamee, P. and H. T Dang. 2009. Overview of the tac 2009 knowledge base population track. In *Proceedings of the Second Text Analysis Conference*.

Rizzo, G., M. van Erp, J. Plu, and R. Troncy. 2016. Making Sense of Microposts (#Microposts2016) Named Entity rEcognition and Linking (NEEL) Challenge. In *6th Workshop on Making Sense of Microposts*, pages 50–59.

Shekarpour, S., K. M. Endris, A. J. Kumar, D. Lukovnikov, K. Singh, H. Thakkar, and C. Lange. 2016. Question answering on linked data: Challenges and future directions. In *WWW 2016*, pages 693–698.

Trivedi, P., G. Maheshwari, M. Dubey, and J. Lehmann. 2017. Lc-quad: A corpus for complex question answering over knowledge graphs. In *International Semantic Web Conference*, pages 210–218. Springer.

Unger, C., C. Forascu, V. Lopez, A.-C. N. Ngomo, E. Cabrio, P. Cimiano, and S. Walter. 2014. Question Answering over Linked Data (QALD-4). sep.

A Appendix 1

Along the paper, we deliver all the necessary scripts to replicate our experiments. There is a README explaining each source code file. All the source code is released under the GPL-v3 license and can be found here.