Knowledge-Based Systems 000 (2017) 1–9

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Grounding proposition stores for question answering over linked data

Bernardo Cabaleiro[a,*], Anselmo Peñas[a], Suresh Manandhar[b]

[a] NLP & IR Group at UNED, C/ Juan del Rosal, Madrid 16 28040, Spain
[b] Department of Computer Science at University of York, Heslington, York YO10 5GH, United Kingdom

ARTICLE INFO

Article history:
Received 17 June 2016
Revised 25 April 2017
Accepted 25 April 2017
Available online xxx

Keywords:
Question answering
Semantic parsing
Linked data
Grounding

ABSTRACT

Grounding natural language utterances into semantic representations is crucial for tasks such as question answering and knowledge base population. However, the importance of the lexicons that are central to this mapping remains unmeasured because question answering systems are evaluated as end-to-end systems.

This article proposes a methodology to enable a standalone evaluation of grounding natural language propositions into semantic relations by fixing all the components of a question answering system other than the lexicon itself. Thus, we can explore different configurations trying to conclude which are the ones that contribute better to improve overall system performance.

Our experiments show that grounding accounts with close to 80% of the system performance without training, whereas training supposes a relative improvement of 7.6%. Finally we show how lexical expansion using external linguistic resources can consistently improve the results from 0.8% up to 2.5%.

1. Introduction

Linked Data refers to *a set of best practices for publishing and connecting structured data on the Web* [1]. It establishes the bases for the Web of Data, an effort from the community of web users to create large amounts structured, machine-friendly knowledge, preserving the structure and semantics of the relations between elements. Although there are plenty of linked data databases (e.g. Freebase [2], DBPedia [3] or Yago2 [4]), common web users lack of the necessary know-how to use them.

Question Answering (QA) can be viewed as one human friendly method for accessing linked data since it alleviates the need to learn query languages such as SPARQL. QA systems typically employ semantic parsing to map natural language into a predicate-argument meaning representation. The map can easily be translated into knowledge base query languages.

We define grounding as the procedure for expressing natural language in terms of the target knowledge base language. More specifically, the task is to map an unbounded number of expressions (natural language) into a small set of entities and properties (linked data). For example the constructions *What does John do for a living?*, *What is John's profession?*, and *Who is John?* are be mapped to the same property `{John - Profession - X}`.

Grounding provides two key benefits. On the one hand, it alleviates the problem of logic form annotation by providing data for indirect supervision [5]. Secondly, if the logic forms share the same vocabulary with the target knowledge base the querying step becomes trivial.

Semantic Parsing methods require a lexicon to enable the mapping between text and the labels of the knowledge base. A *lexicon* captures and ranks the candidate mappings between predicates in natural language and properties in the linked data database. For instance, solving the previous example would require an entry *living* → `profession`. However building these lexicons is not trivial and the contribution to the full system remains unmeasured because the final score is given by the complete system and involves other processes, e.g. choosing the appropriate entry of the lexicon.

Recent work proposes a method to build a lexicon by acquiring knowledge from large text corpora [6]. This process relies on distant supervision to build a lexicon that then is used to fed a semantic parser. Our goal is to study the contribution of this process of knowledge acquisition on closing the gap between natural language and linked data properties. Specifically, it is unclear which syntactic structures should be aligned and what is the impact of each one.

We use our methods of representation and acquisition to transform natural language utterances into logic forms composed by a set of propositions, which are triples with the form

[*] Corresponding author.
E-mail addresses: bcabaleiro@lsi.uned.es (B. Cabaleiro), anselmo@lsi.uned.es (A. Peñas), suresh.manandhar@york.ac.uk (S. Manandhar).

http://dx.doi.org/10.1016/j.knosys.2017.04.016
0950-7051/© 2017 Elsevier B.V. All rights reserved.

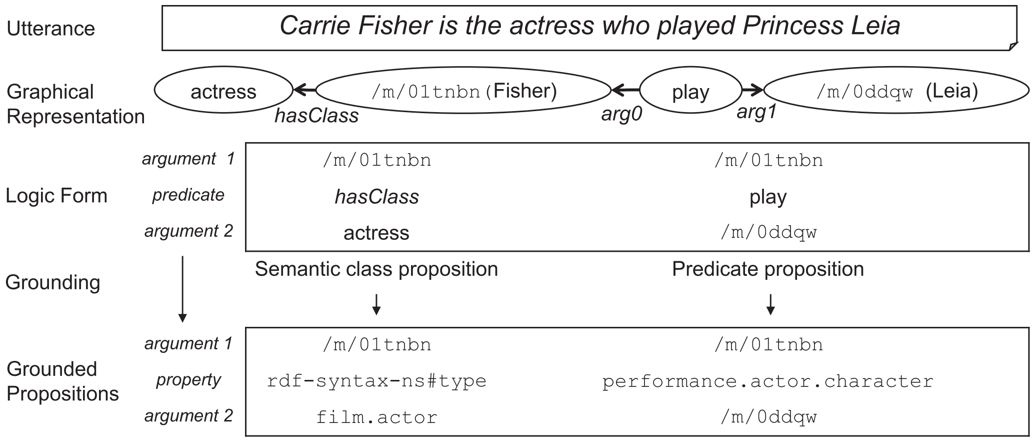

Fig. 1. Example of acquisition of a grounded proposition. For simplicity, we represent the property performance.actor.character as a single triplet. In Freebase, this property is expressed with two triplets related by an intermediate entity.

<argument 1 - predicate - argument 2>. A propositions is mapped into a linked data triple {argument 1 - property - argument 2} to build a grounded proposition, which is a proposition expressed with the linked data vocabulary. We build a lexicon that we denote Grounded Proposition Store (henceforth, GPS) by grounding a large number of propositions automatically extracted from text. Finally, we combine the GPS with the method proposed in [6] to create a scenario where grounding can be evaluated in isolation to study how different grounding configurations affect semantic parsing. Fig. 1 shows how the utterance *"Carrie Fisher is the actress who played Princess Leia"* is transformed to a logic form composed by two propositions and then grounded into linked data properties. We explain this process and some related concepts in Section 4.

We structure our research around the following research questions. In the context of a Semantic Parser trained using raw text for distant supervision:

- *What are the methodological steps to build a GPS?*
- *What is the impact of the GPS when used to fed a semantic parser for question answering?*
- *What linguistic phenomena (syntactic-semantic relations) should be considered in the knowledge acquisition step?*
- *Are external linguistic resources useful for enriching the GPS?*

This article is structured as follows: In Section 2 we motivate the choice of distant supervision using raw text for QA over LD. Section 3 details the architecture of the semantic parser, Section 4 studies the grounding step and presents our approach to build a GPS. In Section 5 we evaluate the effect of GPS in QA over LD and we present the results in Section 6. We finish with some conclusions in Section 7 and propose some future work in Section 8.

2. Semantic parsing over linked data

Early works on semantic parsing for question answering were done on domains with controlled language and small predefined domains such as baseball [7] and geography [8]. However, these approaches cannot be scaled to general-domain knowledge bases.

As semantic parsers scaled to answer a wider range of queries, several problems arise. Firstly, systems have to deal with the lexical variability of the utterances, a problem that grows as domains become less restricted. Secondly, knowledge bases become bigger and richer, so the potential to give wrong answers increases.

Finally, dealing with the variability of knowledge bases also introduces additional challenges since semantic parsers have to adapt to different structures and vocabularies. Currently, many efforts point to linked data databases like DBPedia or Freebase as a source of general domain knowledge. The main reason is that they are a compromise solution between the high quality data that provide the hand-labelled databases and the extension of the automatically generated databases. Linked data databases are often structured in triples that denote relations between two entities, which are named properties. Properties are labelled with a name close to natural language. For example, an instance of the database may be {John - profession - teacher}, although these labels are arbitrary and, in fact, properties are defined extensively by their members.

Early approaches were too dependent on hand-labelled logic forms [8–10], and hence were unable to scale up. More recent work aims to alleviate the supervision problem by using forms of distant supervision, i.e. observation of system behaviour [11], conversations from dialog systems [12], schema matching [13], questions [5] and question-answer pairs [14–18].

GraphParser [6] is a method for distant supervision that hypothesizes that a natural language predicate found in a text expresses a Freebase property. The idea is to identify pairs of entities connected through a predicate in a large document collection and look for the Freebase properties that connect both entities. For example, given the sentence $s =$ *Cameron is the director of Titanic* one of the properties in Freebase between $e_1 = \mathtt{Cameron}$ and $e_2 = \mathtt{Titanic}$ is $r = \mathtt{film.directed\_by}$. Thus, we assume that $\{e_1 - r - e_2\}$ corresponds to the natural language expression *s*.

Distant supervision provides a noisy method to learn weights for each predicate-property pairs. For this purpose, the starting point is to take as prior the frequencies observed in a large text collection to build a lexicon and use it to feed the learning process.

GraphParser tackles this task by pairing reified logic forms derived from a Combinatory Categorial Grammar (CCG) parser [19] with Freebase properties. Each logic form corresponds in turn to a predicate-argument relation. Instead, we show how to obtain similar logic forms from a standard dependency parser. Dependency trees are transformed into graphs, which are then used to extract propositions. Then, propositions are aligned with Freebase to produce a new lexicon.

This setting allows us to measure the effect that different configurations of our Propositions Stores produce on semantic parsing when they are grounded to build the lexicon the system requires.

3. System architecture

In this section we revise the architecture of the question answering system. The system is divided in three main layers: