Answering Natural Language Questions with Intui3

Corina Dima

Seminar für Sprachwissenschaft, University of Tübingen,
Wilhemstr. 19, 72074 Tübingen, Germany
{corina.dima}@uni-tuebingen.de

Abstract. Intui3 is one of the participating systems at the fourth evaluation campaign on multilingual question answering over linked data, QALD4. The system accepts as input a question formulated in natural language (in English), and uses syntactic and semantic information to construct its interpretation with respect to a given database of RDF triples (in this case DBpedia 3.9). The interpretation is mapped to the corresponding SPARQL query, which is then run against a SPARQL endpoint to retrieve the answers to the initial question. Intui3 competed in the challenge called Task 1: Multilingual question answering over linked data, which offered 200 training questions and 50 test questions in 7 different languages. It obtained an F-measure of 0.24 by providing a correct answer to 10 of the test questions and a partial answer to 4 of them.

Keywords: information retrieval, question answering, linked data

1 Introduction

Keyword-based search is the dominant search paradigm today. It permits computers to sift through the massive amounts of unstructured information available on the web and provide the users a ranked list of pages where the target information might be found. However, as remarked in the literature [5], keyword-oriented search has a major drawback: it does not readily provide an *answer*, but *a list of documents* where the answer might be found. The user has to manually inspect each of the provided pages in order to find the actual answer.

The system described in this paper, Intui3, belongs to an alternative search paradigm, that takes a question formulated in natural language and provides a precise answer to it. The system accepts as input a syntactically correct natural language question and constructs its interpretation using syntactic and semantic cues in the question and a target triple store. The construction of a question interpretation is guided by Frege's Principle of Compositionality, namely the interpretation of a complex expression is determined by the interpretation of its constituent expressions and the rules used to combine them. The system uses an ontology, a predicate index and an entity index to construct the interpretations. In the current version all these components are related to the DBpedia 3.9 knowledge base, but with little effort they can be replaced, thus allowing the system to construct interpretations with respect to other knowledge bases.

1.1 Previous work in Question Answering on structured data

Natural language questioning answering systems that serve as an interface to structured databases have a long history, going back to systems like BASEBALL [8] and LUNAR [14] which were able to correctly process and answer natural language questions, but only on a very limited domain. More recent efforts ([12], [16]) have focused on learning to map a complex question to a logical form using an existing lexicon for connecting words to corresponding predicates in a database of facts. These systems focused on the *depth* of the analysis rather than on the *breadth* of the domain. Recently, however, there has been a shift from question answering on small databases to question answering on large knowledge bases ([13], [1]) or to open-domain question answering ([6]). These approaches address the problem of automatically mapping a word sequence to a predicate in the database, as well as the one of constructing and combining possible interpretations. The system Intui3 follows in this trend, as it constructs detailed semantic interpretations for natural language questions by taking into account syntactic and semantic cues in the question and connecting them to the facts available in a specified triple store.

The current paper continues with Section 2 which gives an overview of the main third party resources used by Intui 3. Section 3 details all the steps required to construct a SPARQL interpretation of a natural language question. Section 4 presents the results of Intui3 on the training and test sets provided by the organizers of the QALD4[1] challenge in Task 1, *Multilingual question answering over linked data*. A discussion of issues that affect the interpretation capabilities of the system and possible ways to address them follows in Section 5, which also concludes the paper.

2 Resources

Intui3 makes use of a series of third party resources to construct the interpretation of natural language questions. This section briefly introduces each of them. They are further referenced in Section 3 when describing the interpretation process.

Two natural language processing (NLP) suites are used: SENNA [3] and Stanford CoreNLP [7]. SENNA is a deep neural networks-based system that outputs a host of NLP predictions: part-of-speech (PoS) tags, chunking, name entity recognition (NER), semantic role labeling and syntactic parsing. SENNA's main interest points are its very high processing speed and its state-of-the-art performance. Intui3 uses SENNA (v3.0) for PoS tagging, chunking and performing NER on the input question.

The Stanford CoreNLP suite (v.3.2.0) is used in Intui3 for obtaining lemma information for each token. Stanford CoreNLP is a flexible NLP suite that offers multiple annotators such as PoS taggers, lemmatizers, named entity annotators, sentiment and coreference annotators, as well as annotators for constituency and

[1] QALD4 challenge website: `http://www.sc.cit-ec.uni-bielefeld.de/qald/`

dependency parsing. Multiple annotators can be combined and used to annotate the same input text on various levels.

The system combines the output from the two NLP suites. Initial tests on the QALD4 training set showed that SENNA (v3.0) provides the correct PoS tag labeling of a question more often than Stanford CoreNLP (v.3.2.0) does. SENNA also readily provides the chunking information, which the system uses as a support for constructing the semantic interpretation of the question. SENNA was thus chosen as the main NLP processing suite of the system. SENNA, however, does not offer lemma information, which was then obtained from the Stanford CoreNLP suite.

Intui3 uses a locally installed version of the DBpedia Lookup service[2] [2]. The DBpedia Lookup service provides an easy method for finding DBpedia URIs for a given sequence of words. It is implemented as a Web service and it is based on a Lucene index that provides a weighted label lookup. It combines string similarity with relevance ranking in order to find the most likely matches for a given word or word sequence. To illustrate the advantage of using the DBpedia Lookup service instead of simple string matching techniques we use Q21 from the QALD4 Task 1 test set, *Where was Bach born?* A query for the string *Bach* in the triple store directly would result in a multitude of pages referring either to Bach himself, to his opera or to various other entities that are named after Bach. The first match returned by the DBpedia Lookup service is the URI `http://dbpedia.org/resource/Johann_Sebastian_Bach`, which is, in the case of Q21, the correct choice.

We compiled a list of demonyms and their corresponding country by scraping the information about demonyms that can be found on Wikipedia[3]. The list contains about 600 pairs of the form (demonym, country). This information is also available in DBpedia but the current version (DBpedia 3.9) contains only 440 concepts that are marked as demonym of a country.

The system uses a predicate index obtained by selecting all the predicates in a local DBpedia 3.9 install that was used for obtaining results for the QALD4 challenge. This index contains 49,714 predicates, most of them from the `dbpedia.org/property` namespace (48, 294).

Intui3 computes similarities between words in the question and predicates in the triple store using the word similarity measures implemented in the WordNet Similarity for Java[4] (WS4J) library. All the similarity measures in the package were tested using a set of pairs of the form (word, dbpedia predicate). The focus of this test was to choose a similarity measure that: (i) is able to score pairs of words with different PoS tags; (ii) provides a high score when the word and the predicate are related and a low score when they are not. The measure that fulfilled both criteria was the Hirst & St. Onge similarity measure [9]. It is based on an idea that two lexicalized concepts are semantically close if their WordNet

[2] The code was obtained from `https://github.com/dbpedia/lookup`

[3] List of demonyms scraped from `http://en.wikipedia.org/wiki/Adjectivals_and_demonyms_for_countries_and_nations`

[4] The code for WS4J was obtained from `https://code.google.com/p/ws4j/`

synsets are connected by a path that is not too long and that “does not change direction too often”.

3 Interpreting a Natural Language Question

This section gives a detailed overview of the steps made by Intui3 to interpret a natural language question and to map it to a syntactically correct SPARQL query. The interpretation process involves the following steps: first, the question is tokenized, PoS tagged, and the named entities are identified. Then the question is split into chunks, and the systems assigns initial interpretations to each chunk. The interpretation of the question is constructed by combining the interpretations assigned to each chunk.

We will use question with id 12, Q12: *How many pages does War and Peace have?* from the QALD4 Task1 test set to explain in more detail the interpretation process.

Each question received by the system is pre-processed using a series of standard NLP tools. The SENNA and Stanford CoreNLP suites are used to process the question. The system stores information related to PoS tagging, NER and chunking from the output produced by SENNA and lemma information from the output of Stanford CoreNLP. Next, the system performs a demonym resolution step by looking up each individual token in the question in a list of demonyms and their associated countries (see Section 2 for more details). If a demonym is found, such as the word *German* in the phrase *German lake*, the system retrieves and stores the DBpedia URI identifying the referred country (in this case `http://dbpedia.org/resource/Germany`). Table 1 shows the information obtained in the pre-processing step for Q12. All the information obtained in the pre-processing phase is stored using the tokens as a reference entity.

Table 1. Question pre-processing for *Q12: How many pages does War and Peace have?*

Token	Lemma	PoS	Demonym	NER	SENNA-Chunk	Intui3-Chunk
How	how	WRB	-	O	B-NP	B-WHADJP
many	many	JJ	-	O	I-NP	E-WHADJP
pages	page	NNS	-	O	E-NP	S-NP
does	do	VBZ	-	O	S-VP	S-VP
War	War	NNP	-	B-MISC	B-NP	B-NP
and	and	CC	-	I-MISC	I-NP	I-NP
Peace	Peace	NNP	-	E-MISC	E-NP	E-NP
have	have	VBP	-	O	S-VP	S-VP
?	?	.	-	O	O	O

Intui3 uses sentence chunks as a basis for constructing interpretations. That is why establishing appropriate chunk boundaries is a crucial step towards constructing the correct interpretation. To this end the system combines the chunk information it has obtained from the SENNA chunker and from the SENNA

NER system. Depending on the returned chunks and their associated PoS information, the system can choose to split the provided chunks, or to combine several provided chunks into a larger chunk.

An example of splitting an existing chunk is given in the last column of Table 1, where the chunk *[NP How many pages]* recognized by the chunker is further split into *[WHADJP How many]* and *[NP pages]*.

The converse situation of the system combining multiple chunks can be illustrated using Q42 in the QALD4 Task1 test set, *What is the official color of the University of Oxford?*. The phrase *the University of Oxford* is initially chunked as *[NP the University] [PP of] [NP Oxford]*. The NER system identifies in the same phrase a named entity of type *organization*: *the [ORG University of Oxford]*. Intui3 combines the information from the chunker and the NER system and merges the three initial chunks into a single NP chunk, *[NP the University of Oxford]*.

The processing continues with the interpretation of the chunks obtained in the pre-processing step. The system analyses each chunk individually and assigns one or more interpretations depending on the chunk's type and on the additional semantic and syntactic information available for that chunk. An overview of the types of interpretations that can be assigned by the system is presented below. In all the descriptions, the terms *subject*, *object* and *predicate* refer to the elements of an RDF triple.

- *functional* interpretations define a functor such as *count*, *min* or *max*; such interpretations are triggered by lexical cues (e.g. *how many*).
- *concept* interpretations are used to model class membership via the *rdf:type*[5] property; they are triggered by noun phrases that contain plural nouns like *languages* and are mapped to RDF triples like `(?answer, rdf:type, http://dbpedia.org/ontology/Language)`. The set of reference classes is defined by the DBpedia ontology[6]. The DBpedia ontology defines a hierarchy of 529 classes that include both top-level concepts such as *Person*, *Organization*, *Event*, *Place*, but also more refined concepts such as *Athlete*, *Company*, *SportsEvent* and *Mountain*
- *subject* interpretations are used to map a word sequence to a subject triple as in `(http://dbpedia.org/resource/War_and_Peace, ?p, ?o)`; *object* interpretations are used to map a word sequence to an object triple as in `(?s, ?p, http://dbpedia.org/resource/War_and_Peace)`; such interpretations are triggered by noun phrases such as *War and Peace*, *the official color*, etc.; when a chunk that requires a subject or object interpretation is discovered both subject and object interpretations are generated; this ensures that the system has a chance to investigate all the triples that contain a particular URI in a subject or object position; subject and object interpretations are based on the output returned by the DBpedia Lookup service described in Section 2; if the associated chunk was tagged with the NER label `PER`, the

[5] `http://www.w3.org/1999/02/22-rdf-syntax-ns#type`

[6] `http://wiki.dbpedia.org/Ontology`

system filters the results returned by the DBpedia Lookup service to have the *rdf:type* `dbo:Person`, `foaf:Person` or `yago:Person`. Similarly, the *rdf:type* of the chunks with the NER label `LOC` is restricted to `dbo:Place`. The `MISC` and `ORG` NER labels are too coarse to create such restrictions.

- *empty* interpretations are used to signal the fact that the mapped word sequence does not have a meaning on its own; the triggers for empty interpretations are function words such as *me*, *I*, *do*, *does*, etc.
- *predicate* interpretations are used to map a word or word sequence in the question to a predicate that is available in the triple store; predicate interpretations are triggered either by verbs (e.g. *spoken*), common nouns (e.g. *pages*, *ingredients*) or noun phrases (e.g. *programming languages*, *the official color*); as opposed to subject, object or concept interpretations, the predicate interpretations are not immediately resolved, as their interpretation is highly dependent on the exact question context; instead, the predicate interpretation stores a *predicate pattern* that is taken into account in the interpretation process;
- *triple* interpretations are used to map a sequence of words to an RDF triple; *subject*, *object* and *concept* interpretations are all instances of triple interpretations;
- *query* interpretations are used to map a sequence of words to a SPARQL query; query interpretations are the most complex type of interpretation that can be constructed by the system, and are obtained through composition from the other types of interpretations

Each type of interpretation comes with a set of combination rules. These rules define how the current interpretation can be combined with any other type of interpretation and also construct the combined interpretation.

Table 2 presents all the interpretations that have been initially assigned to the chunks in our running example, *Q12: How many pages does War and Peace have?* The next step involves traversing all the chunks in a right-to-left order and combining the interpretations.

$$I_{Q12} = [I_{how\_many}, I_{pages}, I_{does}, I_{War\_and\_Peace}, I_{have}] \tag{1}$$

First, the interpretations of rightmost two chunks ($I_{War\_and\_Peace}$ and I_{have}) are combined by considering each possible pair of interpretations of the two chunks in the order they appear in the sentence. The possible number of interpretations is the product of the number of interpretations for each chunk. In our case, $|I_{War\_and\_Peace}| = 10$ and $|I_{have}| = 1$, so there are $10 * 1 = 10$ possible combinations. In this particular case, I_{have} contains only the empty interpretation, so the interpretations that result by combining $I_{War\_and\_Peace}$ and I_{have} are the interpretations in $I_{War\_and\_Peace}$.

For each subsequent step we combine the results of the previous combination with the rightmost interpretation that has not yet been combined. In our case, we combine I_{does} with the result of combining $I_{War\_and\_Peace}$ and I_{have}, which was $I_{War\_and\_Peace}$. I_{does} contains again only the empty interpretation, so the result of this combination is still only $I_{War\_and\_Peace}$.

In the next step we combine I_{pages} with the results of the previous combinations, in our case $I_{War\_and\_Peace}$. This time we have $|I_{pages}| * |I_{War\_and\_Peace}| =$ 20 combined interpretations. The combination of a subject and a predicate interpretation involves querying the triple store for all the triples with the given subject and extracting a set of predicates that co-occurred with the subject. All the extracted predicates in this set are scored with respect to the pattern provided in the predicate interpretation. In our example, the system looks for all the triples with the subject `http://dbpedia.org/resource/War_and_Peace`, extracts the set of predicates it occurs with (25 different predicates), and scores all of them with respect to the predicate pattern (*page* or *pages*).

The scoring mechanism uses two scorers chained together: first a string similarity scorer, then a WordNet-based similarity scorer which uses the Hirst-St.Onge similarity measure (see Section 2). Chaining allows the system to look first for lexical similarities between the candidate predicates and the provided pattern, and if the lexical similarity is too low, to look for similarity at the semantic level. This allows the system to find both the lexical similarity between the predicate `http://dbpedia.org/property/pages` and the pattern *pages* and the semantic similarity between `http://dbpedia.org/property/spouse` and the pattern *husband*. The system collects all the candidate predicates that score above a specified threshold and constructs query interpretations by combining the predicate interpretations for I_{pages} with the subject/object interpretations for $I_{War\_and\_Peace}$. 13 query interpretations are constructed in this particular case, one of them displayed in Listing 1.1.

Listing 1.1. Final query for Q12: *How many pages does War and Peace have?*

```
SELECT DISTINCT ?answer
WHERE
{
      <http://dbpedia.org/resource/War_and_Peace>
      <http://dbpedia.org/property/pages>
      ?answer .
}
```

All the combined interpretations are scored by multiplying the scores of the initial interpretations. The last step made by the system in the particular case of Q12 is combining the interpretations it has obtained until this point with I_{how_many}. The *functional:count* interpretation can only be combined with a query interpretation and requires a numeric answer. To obtain the combined interpretation, the system first runs the query interpretation and retrieves the answers. If there is only one answer that can be cast to a number, then the combined interpretation will be the existing query interpretation. Otherwise, the system constructs the SPARQL representation of the combined interpretation by attaching a count clause to the existing query.

The system outputs the *query* interpretation with the highest score as the final interpretation, or OUT OF SCOPE if the final interpretation is an *empty* interpretation. If the final interpretation is any of the other types of interpretation, then the system cannot construct a valid interpretation for that particular question and thus cannot answer the question.

In the case of Q12: *How many pages does War and Peace have?*, the system chooses the query in Listing 1.1, which provides the correct answer, *1225* pages.

Table 2. Interpretations for *Q12: How many pages does War and Peace have?* The question mark does not have an interpretation as it was not assigned to a chunk.

Phrase	Phrase Type	Interpretation
How many	WHADJP	Functional:COUNT
pages	NP	Predicate: `pattern`=*page* Predicate: `pattern`=*pages*
does	VP	Empty
War and Peace	NP	Subject [`http://dbpedia.org/resource/War_and_Peace`: 0.70] Subject [`http://dbpedia.org/resource/War_and_Peace_(opera)`: 0.33] Subject [`http://dbpedia.org/resource/Paris_Peace_Conference,_1919`: 0.29] Subject [`http://dbpedia.org/resource/Peace_and_conflict_studies`: 0.23] Subject [`http://dbpedia.org/resource/Peace_movement`: 0.20] Object [`http://dbpedia.org/resource/War_and_Peace`: 0.70] Object [`http://dbpedia.org/resource/War_and_Peace_(opera)`: 0.33] Object [`http://dbpedia.org/resource/Paris_Peace_Conference,_1919`: 0.29] Object [`http://dbpedia.org/resource/Peace_and_conflict_studies`: 0.23] Object [`http://dbpedia.org/resource/Peace_movement`: 0.20]
have	VP	Empty

4 System Results in QALD4: Task1

Intui3 was evaluated on the training and test sets offered by the organizers of QALD4 in Task 1, *Multilingual question answering over linked data*. Although the organizers offer questions in seven languages, Intui3 can only interpret questions written in English.

The results of the system on the training and the test set are summarized in Table 3. Table 4 presents the questions answered correctly or partially correct by Intui3, as well as their individual scores.

Table 3. Intui3 results on the QALD4 Task 1 training and test sets. The reported recall, precision and F-score refer to the results obtained by the system on the whole test/training set.

Dataset	Size	Constructed queries	Correct	Partially correct	Recall	Precision	F-measure
QALD4 test	50	33	10	4	0.24	0.23	0.24
QALD4 train	200	137	38	5	0.20	0.20	0.20

Table 4. Test questions answered correctly or partially correct by Intui3.

ID	Question	Precision	Recall	F-measure
4	How many programming languages are there?	1.0	1.0	1.0
5	Give me all Dutch parties.	1.0	1.0	1.0
8	Which rivers flow into a German lake?	1.0	0.87	0.93
12	How many pages does War and Peace have?	1.0	1.0	1.0
13	Which ingredients do I need for carrot cake?	1.0	1.0	1.0
15	Who has Tom Cruise been married to?	1.0	1.0	1.0
18	Give me all films produced by Steven Spielberg with a budget of at least $80 million.	0.33	0.06	0.1
29	Give me all Australian metalcore bands.	0.61	0.01	0.02
31	How many inhabitants does the largest city in Canada have?	1.0	1.0	1.0
32	In which countries can you pay using the West African CFA franc?	0.5	0.7	0.6
35	To which artistic movement did the painter of The Three Dancers belong?	1.0	1.0	1.0
36	Which pope succeeded John Paul II?	1.0	1.0	1.0
42	What is the official color of the University of Oxford?	1.0	1.0	1.0
50	How many gold medals did Michael Phelps win at the 2008 Olympics?	1.0	1.0	1.0

5 Discussion and Future Work

There are several types of issues that prevented the system from having a better coverage with respect to the provided datasets. We list them below, together with possible methods for alleviating them:

- The system relies heavily on the part-of-speech tagging information when it chooses a possible interpretation for a chunk. The prediction of a part-of-speech tagger can, however, be incorrect, especially in the case of questions. For example in the case of Q7, *Does the Isar flow into a German lake?* the part-of-speech tagger assigned the label NN to the verb *flow*, thus preventing a correct chunking (*[the Isar flow]* was analyzed as one NP chunk). Such mistakes might seem unlikely from a system like SENNA that reports an per-word accuracy of 97.29%. But, as noted in [11], although PoS tagging is considered a “solved” problem due to systems that report over 97% *per-word* accuracy, the same systems only achieve 55 to 57% *per-sentence* accuracy. A better performance on PoS tagging questions might be obtained by re-training the PoS tagger using both question and non-question data. Such approaches have been shown to significantly improve parsing performance on question data [10].
- The purely right-to-left method of combining the interpretations does not always provide the best method of constructing the question's interpretation. For example in the case of Q11, *Give me all animals that are extinct*, the interpretation should start from the left rather than from the right. The motivation for combining the interpretations using the such heuristics instead of the output of a parser is given by our previous experience in the QALD3 challenge. There our system [4] used the output of a constituency parser to construct interpretations. However, as the PoS tags were often incorrect, the

parser would have no possibility to construct the correct parse and the system would fail to provide any interpretations. The SENNA chunker, on the other hand, provided to be more robust: in some cases the system generated an incorrect PoS tag for a word in the chunk, but the chunk itself was correctly labeled. The heuristic for combining interpretations can be improved by constructing interpretations in both directions and choosing the overall best scoring one. Another avenue worth investigating is to use the output of a dependency parser as a method for choosing the order in which the chunk interpretations should be combined.

- In the current system the algorithm for refining chunk boundaries uses handwritten rules; a better solution would be to develop a dedicated chunker that can provide the correct chunk boundaries for constructing interpretations.
- The test set included many questions that have noun phrases containing superlative adjectives (e.g. *the tallest player*, *the most books*, *the youngest Darts player*, etc.). The current system is not equipped to construct an interpretation for such cases. Such phrases are correctly interpreted only if a dedicated predicate for that phrase exists in the knowledge base (e.g. the predicate *largestCity* for an entity of type *country* in Q31).
- The system is not currently equipped to handle questions that require a Yes/No answer.

Intui3 is restricted to answering questions in English. Porting the system to work with other languages is possible, although restricted to those languages that have all the NLP tools required by Intui3 (PoS tagger, NER system, chunker, lemmatizer, a wordnet and the corresponding similarity measures). Another necessary resource is an instance of the DBpedia Lookup service customized for the target language.

We plan to further improve the system by taking into account all the issues presented above. Another goal is to test the adaptability of the system by trying to answer questions with respect to other freely available knowledge bases, using already existing sets of questions and answers like the ones described in [1] and [15].

Acknowledgments. The research leading to these results has received funding from the German Research Foundation (DFG) as part of the Collaborative Research Center 'Emergence of Meaning' (SFB 833). The author would like to thank Emanuel Dima for his comments on the initial version of the paper, as well as the anonymous reviewers for their suggestions.

References

1. Berant, J., Chou, A., Frostig, R., Liang, P.: Semantic parsing on freebase from question-answer pairs. In: Proceedings of EMNLP (2013)
2. Bizer, C., Lehmann, J., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R., Hellmann, S.: Dbpedia-a crystallization point for the web of data. Web Semantics: Science, Services and Agents on the World Wide Web 7(3), 154–165 (2009)

3. Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., Kuksa, P.: Natural language processing (almost) from scratch. The Journal of Machine Learning Research 12, 2493–2537 (2011)
4. Dima, C.: Intui2: A prototype system for question answering over linked data. Proceedings of the Question Answering over Linked Data lab (QALD-3) at CLEF (2013)
5. Etzioni, O.: Search needs a shake-up. Nature 476(7358), 25–26 (2011)
6. Fader, A., Zettlemoyer, L.S., Etzioni, O.: Paraphrase-driven learning for open question answering. In: ACL (1). pp. 1608–1618 (2013)
7. Finkel, J.R., Grenager, T., Manning, C.: Incorporating non-local information into information extraction systems by gibbs sampling. In: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics. pp. 363–370. Association for Computational Linguistics (2005)
8. Green Jr, B.F., Wolf, A.K., Chomsky, C., Laughery, K.: Baseball: an automatic question-answerer. In: Papers presented at the May 9-11, 1961, western joint IRE-AIEE-ACM computer conference. pp. 219–224. ACM (1961)
9. Hirst, G., St-Onge, D.: Lexical chains as representations of context for the detection and correction of malapropisms. WordNet: An electronic lexical database 305, 305–332 (1998)
10. Judge, J., Cahill, A., Van Genabith, J.: Questionbank: Creating a corpus of parse-annotated questions. In: Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics. pp. 497–504. Association for Computational Linguistics (2006)
11. Manning, C.D.: Part-of-speech tagging from 97% to 100%: is it time for some linguistics? In: Computational Linguistics and Intelligent Text Processing, pp. 171–189. Springer (2011)
12. Tang, L.R., Mooney, R.J.: Using multiple clause constructors in inductive logic programming for semantic parsing. In: Machine Learning: ECML 2001, pp. 466–477. Springer (2001)
13. Unger, C., Bühmann, L., Lehmann, J., Ngonga Ngomo, A.C., Gerber, D., Cimiano, P.: Template-based question answering over rdf data. In: Proceedings of the 21st international conference on World Wide Web. pp. 639–648. ACM (2012)
14. Woods, W.A.: Progress in natural language understanding: an application to lunar geology. In: Proceedings of the June 4-8, 1973, national computer conference and exposition. pp. 441–450. ACM (1973)
15. Zelle, J.M., Mooney, R.J.: Learning to parse database queries using inductive logic programming. In: Proceedings of the National Conference on Artificial Intelligence. pp. 1050–1055 (1996)
16. Zettlemoyer, L.S., Collins, M.: Online learning of relaxed ccg grammars for parsing to logical form. In: In Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL-2007. Citeseer (2007)