Definitional Question Answering Using Text Triplets

Chandan Kumar, Ch. Ram Anirudh and Kavi Narayana Murthy

Abstract Definitional question answering deals with answering questions of the type "Who is X" and "What is X." The techniques used in the literature extract long sentences that may not only give irrelevant facts, but also pose difficulty in evaluating the performance of the system. In this paper, we propose a technique that uses text triplets. We further choose relevant triplets based on a manually built list of terms that are found in definitions in general. The selected triplets give simple, short, and precise definitions of the target. We also show that evaluation becomes easy.

Keywords Question answering · Definitional question answering · Triplets

1 Introduction

Finding answers to arbitrary questions is a highly complex task. Most question answering (QA) systems restrict themselves to searching for sentences in a given corpus and possibly selecting parts of these sentences, which are intended to contain the answers sought by the users.

The text retrieval conference (TREC) series introduced a number of text processing tasks in a formal way and participating groups worked on these specified tasks in a competitive spirit. The QA task was first introduced in 1999 [1]. The task of QA is to find answers in a collection of unstructured text to questions posed in natural language. Initially, only factoid and list questions were considered. An answer to a factoid question is generally a short span of text, i.e., a word, a phrase

C. Kumar (✉) · Ch. R. Anirudh · K. N. Murthy
School of Computer and Information Sciences, University of Hyderabad, Hyderabad, India
e-mail: chandanmcmi20@gmail.com

Ch. R. Anirudh
e-mail: ramanirudh28@gmail.com

K. N. Murthy
e-mail: knmuh@yahoo.com

K. S. Raju et al. (eds.), *Data Engineering and Communication Technology*,
Advances in Intelligent Systems and Computing 1079,
https://doi.org/10.1007/978-981-15-1097-7_10

(e.g., How many calories are there in a Big Mac? What is the capital city of India?). List Questions ask for list of items (e.g., List all states in India, List all Universities in India). TREC first introduced and defined definitional question answering in its 2003 edition. Definitional questions are questions such as *What is X? Who is X?*, henceforth termed as type 1 and type 2 question, respectively. An answer to a definitional question should be a collection of facts that define the term being questioned as precisely as possible. Types of facts that define the questioned terms vary from term to term. The challenge is to find suitable facts for a given term. Note that no attempt is made to define a term precisely, that is an extremely hard problem, if not impossible. The goal is only to find parts of a given corpus which can help someone in getting some idea of what or who X is.

These Definitional QA systems suffer from a lack of standard representation of the answer. Either whole sentences are given out or parts of sentences selected somewhat arbitrarily are given out as answers. Length of answers varies quite a bit. All these make the task of evaluating the precision of answers very difficult [2]. These are the main issues addressed in this paper.

2 Related Work

Many groups participated in TREC 2003. All the groups went through the same pipeline of processes: question processing, information retrieval (IR) to find relevant documents from the document collection, candidate sentence selection, sentence ranking, and redundancy removal [1]. Most of the groups used their own IR engines and retrieved relevant documents from the AQUAINT [3] corpus, taking it as the source of the answer. TREC also provided a collection of relevant documents to 50 selected definitional questions. The participating groups applied a variety of heuristics to select candidate sentences. Word overlapping measure and text summarization techniques were used by participating groups for redundancy removal.

BBN (Bolt, Beranek, and Newman) [4] defined kernel facts as phrases extracted from a candidate sentence in a specific way. They defined four types of kernel facts: appositives and copula, propositions, structured pattern, and relation. They extracted these kernel facts using linguistic processing with the help of an information extraction (IE) tool. They used full candidate sentences if kernel facts of the above types could not be extracted. They ranked the extracted kernel facts by calculating the similarity measure to the profile of the question [4]. BBN got the highest *F* measure score of 0.55.

Qualifier [5] is a question answering system developed at the National University of Singapore. It applied co-reference resolution to relevant documents returned by an IR tool. It put all sentences which contain any part of the question target in the positive set and other sentences in the negative set. This system ranked the candidate sentences (positive set) two times. First, it ranked the sentences statistically. A sentence is scored by using a combination of scores of each word present in the sentence, and the score of a word is being computed from its frequency. Candidate sentences were then ranked again using a repository of definitional patterns. Finally, the qualifier system applied an MMR text summarization technique to eliminate redundancy and produced the final answer. Qualifier got the second highest F measure score of 0.47.

TextMap [6], an NLP group at the University of Southern California, differs from other works only in the ranking techniques used. This group used four resources to rank the candidate sentence: a collection of biographies, a collection of descriptors of proper people, wordnet, and semantic relationship patterns. This group generated variable-length answers and got the third highest F measure score of 0.46.

The MIT group [7] developed three modules: database lookup, dictionary lookup, and document lookup. Each module generated an answer to a definitional question, and a final answer was generated by merging the answers from all the modules. In the database lookup module, a database was built by applying 13 surface patterns to the Acquaint corpus offline and answer of a question was found by querying the question target in this database. In the dictionary lookup module, answer projection technique was applied to map answer to a question from the dictionary to the corpus. In the document lookup module, sentences containing question targets from the relevant documents returned by IR were returned as the answer. This module comes into action if the first two modules fail. MIT also generated variable-length answers. It got an *F* measure score of 0.30. An extension of the work by the same team presented a component-level evaluation of each module [8].

Han et al. [9] used a probabilistic model consisting of three parameters: a topic model, definition model, and sentence (language) model. The goal is to find the probability that a sentence is a definition given a topic (target) ($P(D, S|T)$). Cui et al. [10, 11] explored probabilistic lexico-syntactic pattern matching, also known as soft pattern matching models, for answer extraction. Chen et al. [12] used *N*-gram language models for re-ranking the answers extracted. Paşca et al. [13] extracted answers from text snippets extracted from web that are anchored with time information. The idea is that these texts inform about an event associated with the target.

TREC 2003 used only the second pass of evaluation technique of “The Definitional Pilot,” a series of the pilot evaluations as part of the AQUAINT program [2]. This aspect is discussed in detail in a later section.

3 Key Observations

Current systems either throw out entire sentences or some parts of selected sentences. Answers are thus variable-length text strings, even the syntactic structure may vary significantly from item to item. When lengthy sentences are given out as answers, parts may be irrelevant or distractive. For example, for ‘Sunderbans,’ the retrieved answers could be:

`The Sunderbans in West Bengal and the Gahirmatha coast in Orissa are what could be called the stars of mainland India’s coastal and marine ecosystems.`

`In the mangrove forests of Sunderbans, West Bengal, Ramakrishna Mission Lokasiksha Parishad (RKMLSP) and Sri Ramakrishna Ashram Nimpith (SRAN) are two organizations that help local communities understand and overcome problems arising from their unique surroundings.`

It is easy to see that the second sentence has many parts which are not relevant or useful for defining what ‘Sunderbans’ are.

Existing systems also use varied resources to rank candidate sentences: word vectors, dictionaries, collection of biographies, wordnet glosses, semantic relationship patterns, etc. [14]. It would be good if we can minimize and standardize the use of external resources for system development as also for evaluation.

3.1 Problems of Evaluation

Drawing from [2], consider the question “Who is Christopher Reeve?” List of concepts extracted by human experts for the purposes of evaluation may be

1. Actor
2. Accident
3. Treatment/Therapy
4. Spinal cord injury
5. Activist
6. Written an autobiography
7. Human embryo research activist

Let us say the response from system is

1. Actor
2. The actor who was paralyzed when he fell off his horse
3. The name attraction
4. Stars on Sunday in ABCs remake of rear window
5. Was injured in a show jumping accident and has become a spokesman for the cause

How do we now count the matching concepts and calculate precision and recall? String matching will not work. Earlier researchers have manually marked the concepts and tried to match. For example, *paralyzed when he fell off his horse* is taken as a concept and is manually equated to the concept *accident*. Thus, counting the total number of facts in answer string becomes subjective and hard if lengthy and verbose answers are generated. It would be good if answers always conform to a specified structure. For example, going back to the question target "Sunderbans", if the system could generate

```
   The Sunderbans in West Bengal and the Gahirmatha coast in
Orissa
   mainland India's coastal and marine ecosystems
   mangrove forests of Sunderbans
```

that would be so much better both as an answer to the given question and for evaluation by comparing with reference answers.

The 'definitional pilot' was the first of the series of pilot evaluations for question answering where the objective of each pilot was to come with effective evaluation technique for certain type of questions [2]. The definitional pilot was completed in two rounds by two human assessors. In each round, both the assessors evaluated and ranked eight runs submitted by different participating groups. In the first round, each run was evaluated by two scores, one for the content of answer and one for the order of the answer. Two rankings of the eight runs by two different human assessors in the first round varied a lot due to the order score. In the second round of evaluation, the system runs were evaluated by only the content score. This time the two rankings of the system runs were more similar to each other. Thus, more stable evaluation technique was found. In the second round, the assessor made a list of nuggets by reading the system responses and classified nugget as vital or non-vital. An information nugget was defined as a fact for which the assessor could make a binary decision as to whether a response contained the nugget. Vital facts are essential to make a definition good, whereas non-vital facts act as *do not care* (they should not be awarded or penalized). The content score was computed by F-measure, a combination of R (Recall) and P (Precision).

$$\text{Recall} = \frac{\text{number\_retrieved\_vital}}{\text{total\_number\_vital\_on\_list}} \tag{1}$$

$$\text{Precision} = \frac{\text{number\_retrieved\_relevant}}{\text{total\_number\_retrieved}} \tag{2}$$

$$F_{\beta} = \frac{\left(\beta^2 + 1\right) \times \text{Precision} \times \text{Recall}}{\beta^2 \times \text{Precision} + \text{Recall}} \tag{3}$$

Calculating recall was straightforward, but this was not the case with precision. It is not easy to say how many facts are present in a sentence because strings can be substrings of other strings. A trial evaluation before the pilot showed that assessors found enumerating all concepts represented in a response to be so difficult as to be unworkable. For example, how many concepts are contained in "stars on Sunday in ABC's remake of Rear Window"? Very long answers need to be penalized in some way. A crude approximation of precision was made by giving an allowance of 100 characters for each fact. The precision was downgraded proportionately for answers longer than this allowance [2].

4 The Proposed System

We represent all text in the form of triplets. A text triplet is a three-tuple: (subject; relation; object). Here, subject and object are some entities and relation is a relationship between these two entities. This is similar to the RDF framework. It must be noted that the terms subject and object do not necessarily conform to the linguistic notions of subject and object.

We use the Stanford Open Information Extraction tool to generate text triplets for a given sentence. Text triplet representation of the answers increases the precision of the answers—we can hopefully retain only the relevant parts of the candidate sentences. This also makes evaluation easier—it is easier to check if specific triplets are found in the reference answers or not.

4.1 System Architecture

The proposed system architecture is shown in Fig. 1. We could simply do a regular expression-based search in the text corpus for sentences containing the target word and process only these sentences further. However, we see that the Stanford Open IE tool, which we use later to extract triplets, is capable of co-reference resolution. Thus, sentences, not containing the target word but containing pronouns that refer to the target word, can also be included. This requires that entire documents are selected at this stage, not just the sentences containing the target words. The following example illustrates how the Stanford Open IE tool handles co-references:

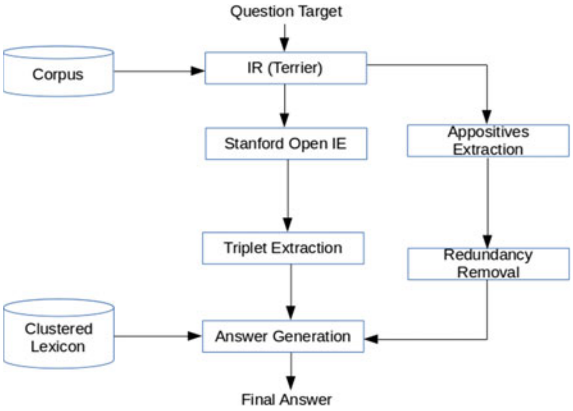
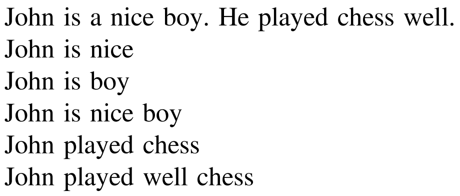


Fig. 1 Proposed system architecture



John is a nice boy. He played chess well.
John is nice
John is boy
John is nice boy
John played chess
John played well chess

It may be observed that the pronoun ‘he’ has been replaced with ‘John.’

An information retrieval (IR) tool is used to extract documents that are relevant for the question target, which is treated as the query string for IR. Terrier [15], an open-source search engine, readily deployable on large-scale collections of documents, is used here. The IR tool is first trained on a large collection of text documents before using it. The IR tool returns a list of documents which are hopefully relevant for the given query string, along with a score. Only the top 30 documents are retained for further processing.

Then, the documents returned by the above module are given to Stanford Open IE tool as input, and text triplets are generated in this module. All the text triplets generated by the Stanford Open IE tool are not equally useful. We select only the triplets which contain question target as the subject.

Stanford Open IE tool Stanford Open IE [16] is a tool developed at Stanford University for the task of open domain information extraction. Open domain information extraction deals with identifying all the entities and relations among them present in text and extracting them. The tool first breaks a sentence into some entailed clauses. These entailed clauses are maximally shortened further, and text triplets are generated. Text triplet is a three tuple <*subject*; *relation*; *object*> where subject and object are entities, and relation is binary relation between these two entities.

Triplets generated by Stanford Open IE tool for the sentence “Abdul Kalam better known as A. P. J. Abdul Kalam (October 15, 1931 to July 27, 2015) was the 11th President of India from 2002 to 2007” are shown in Table 1. It may be observed that the text triplets 7, 8, 9, 10, and 11 are redundant, given the text triplet 6.

Answer Extraction: A good definition should include all the relevant points, must avoid or at least minimize redundancy. Therefore, simply giving out whatever triplets match the target word is not a good idea. As we have seen above, this can lead to a lot of redundancy. Also, we need to handle synonyms and equivalent terms in order to increase the recall. In this paper, we propose a method to address these issues. We develop and use what we call ‘clustered lexicons.’ Clustered lexicons are collections of synonyms, morphological variants, and equivalent expressions in general. For example, *originated*, *located*, *situated*, *lies* may all denote location of an organization. As an example, the Stanford Open IE tool may generate a triplet containing ‘located’ as a relation, and we can expand this to include triplets containing ‘situated,’ etc. More importantly, if we are looking for answer to a question of the type ‘who is X?,’ we know what kinds of information

Table 1 Triplets generated by StanfordOpenIE for sentence "Abdul Kalam better known as A. P. J. Abdul Kalam (15 October 1931 27 July 2015) was the 11th President of India from 2002 to 2007."

No	Subject	Relation	Object
1	Abdul Kalam	known as	15 October 1931
2	Abdul Kalam	known as	A. P. J. Abdul Kalam
3	Abdul Kalam	was President from	2002–2007
4	Abdul Kalam	was	President
5	Abdul Kalam	better known as	A. P. J. Abdul Kalam
6	Abdul Kalam	was	11th President of India from 2002 to 2007
7	Abdul Kalam	was 11th President of	India
8	Abdul Kalam	was President of	India
9	Abdul Kalam	was 11th President from	2002 to 2007
10	Abdul Kalam	was	11th President
11	Abdul Kalam	was	President of India from 2002 to 2007
12	Abdul Kalam	better known as	15 October 1931

we want in the answers. We may want to know his place and date of birth, his affiliation and position or status, his achievements, his contributions, etc. So, instead of directly working with all the large number of triplets we may have obtained, we should instead start from what kinds of information we wish to get and try and locate such triplets. We can easily avoid redundancy, and we can include only one triplet which talks about the place of birth, for example. This way, we can seek and include whatever information we need in the final answer, avoiding redundancy and including equivalent expressions as needed to enhance recall.

The clustered lexicons were developed using the same Stanford Open IE tool. We first collect certain categories of terms and put them in different clusters. For example, we put *Economics, Business, Finance, Banking, Import, and Export* in a separate cluster for type 1 questions. Similarly, we put *Scientist, Inventor, Engineer, and Subject Expert* in a separate cluster for type 2 questions. We choose these terms manually look up the definitions of the terms in a dictionary, and these definitions are given to the Stanford Open IE tool for triplet extraction. The triplets so generated are used to further enhance these clustered lexicons. The Stanford Open IE tool may give relations such as *known as, also known as, better known as*. We cut down on redundancy and retain only the term 'known' in the clustered lexicon. The clustered lexicons include nouns, verbs, and adverbs.

Appositives are noun phrases that define their adjacent nouns and occur frequently in news articles. They are useful in definitional question answering. For example, in the sentence "*The trees, some of which grow only in the Sundarbans, the world's largest mangrove swamp, were felled to pave the way for fisheries*," appositive is, "*Sundarbans, the world's largest mangrove swamp*." Stanford Open IE tool is not able to recognize the appositives. We use a natural language Parser to extract appositives. The PCFG parser [17], which is one of the six parsers in the Stanford Lexicalized Parser v3.9.1, is used here. The parser tags appositives with the ‘appos’ tag. Some appositives may occur in multiple documents. We retain only the unique set of appositives.

The final list of triplets is obtained by expanding the triplets given by the Stanford Open IE tool using the clustered lexicons, and adding the appositives extracted separately. There is a need for some normalization of the text triplets. Triplets are sorted on the basis of the number of words they contain. If all words of the smaller triplet occur in a bigger triplet, then the smaller triplet is considered duplicate and eliminated. Triplets that contain the same subject and the same relation, differing only in objects, are merged.

4.2 Improvements to Evaluation

The problem with calculating precision is counting the total number of facts in the answer string. Text triplet representation of the answers makes this simple. The total number of facts present in the answer string can be taken as the total number of objects in the text triplets. For example, consider three text triplets (Gandhi; was born; on October 2, 1869) (Gandhi; was born; at Porbandar) (Gandhi; was born; as Mohan Das). Here, the total number of facts = the total number of the objects = 3 (namely, on October 2, 1869, at Porbandar and as Mohan Das). Now, recall and precision are calculated as

$$\text{Recall} = \frac{\text{number\_retrieved\_vital\_objects}}{\text{total\_number\_vital\_objects\_on\_list}} \tag{4}$$

$$\text{Precision} = \frac{\text{number\_retrieved\_relevant\_objects}}{\text{total\_number\_retrieved\_objects}} \tag{5}$$

5 Experiments and Results

A collection of news articles from *The Hindu* daily newspaper published during the period 2006–2010 is used here in our experiments.

Clustered lexicons were developed as described above. We choose 70 terms for type 1 and 20 terms for type 2 questions, and we collected definitions of these terms manually. These definitions were given to the Stanford open IE tool as input, and clustered lexicons were built by manual inspection and careful selection. Clustered lexicons for type 1 and type 2 questions contained 267 and 76 words, respectively.

We prepared 50 test questions selecting the terms manually, 30 for type 1 and 20 for type 2. Making a list of all the nuggets from a large collection of the documents is tedious. To make this task easy, we used another source of answers, namely Wikipedia. Now, we could do two different experiments. We collected Wikipedia

Table 2 Test Results

	What			Who			Overall		
	R	*P*	*F*	*R*	*P*	*F*	*R*	*P*	*F*
Wikipedia	0.65	0.85	0.65	0.52	0.89	0.52	0.58	0.87	0.58
The Hindu	0.44	0.80	0.44	0.29	0.83	0.29	0.39	0.81	0.39

pages for 50 question targets and made a final list of facts by reading each Wikipedia page for each question target. Further, we classified nuggets as vital or non-vital. We evaluated the system responses generated by our system from Wikipedia pages. The content score is measured by F measure. For the second experiment, we first run our system on the collection of the Hindu news articles and collect system responses. We made a list of nuggets by reading system responses. We calculated the average number of facts from nuggets list of Wikipedia. Later, we used this number as the number of retrieved nuggets (denominator in the *precision*), if the number of nuggets retrieved in the response was less than the average number of the Wikipedia nuggets.

For the key word ‘Sundarbans,’ the Terrier tool retrieved 134 documents. Top 30 of these were considered for further processing by the Stanford Open IE Tool for triplet extraction and Stanford Parser for extracting appositives. Stanford Open IE generated 1309 triplets, of which 179 had the key term ‘Sundarbans.’ Stanford parser generated 50 appositives, of which 7 had ‘Sundarbans.’

From Table 2, it can be seen that when tested on news articles and Wikipedia, our system could produce a precision of 0.81 and 0.87, respectively. Wikipedia generally contains more of definitional facts than news articles. Also, short answers such as these would be generally more preferable to lengthy sentences, parts of which may be completely unrelated. See Tables 3, 4, and 5 for answers retrieved by the system.

Table 3 Results for ‘Sundarbans’—The Hindu

Cluster No.	Text triplets
1	Sundarbans; are popularly referred to; to last refuge of tiger
3	Sundarbans; are; popularly referred
3	Sundarbans; is in; South 24 Parganas district of West Bengal
3	Sundarbans; is home to; estimated 425 species of wildlife including 300 species of birds
8	Sundarbans; has; has long declared, has battered by rains caused by deep depression in last two days

Table 4 Results (appositives) for 'Sundarbans'—The Hindu

Appositives
appos(Sundarbans-11, swamp-18)
appos(Aila-4, Sundarbans-9)
appos(Aila-4, Sundarbans-9)
appos(Sundarbans-7, landscape-11)
appos(Sundarbans-25, forest-33)
appos(Sundarbans-13, delta-18)
appos(Sundarbans-25, forest-33)

Table 5 Results for 'Sundarbans'—Wikipedia

Cl. No.	Text triplets
3	Sundarbans; is network of; marine streams
3	Sundarbans; is vast forest in; coastal region of Bay of Bengal
8	Sundarbans; contain; world's largest coastal mangrove forest with area

6 Conclusions

In this paper, we have presented our preliminary work on question answering for 'Who is X' and 'What is X' kinds of questions. We believe working with short and structured pieces of texts, such as the triplets we have described, would be better than working with lengthy sentences. We have shown how we can possibly extract short, crisp, and more relevant and precise answers to definitional questions. Evaluation also becomes easier.

Working with triplets may be better not only for the present task but for many other tasks in Natural Language Processing. More rigorous studies are needed to firmly establish this.

References

1. Voorhees, E.M., Dang H.T.: Overview of the TREC 2003 question answering track. In: TREC 2003, pp. 54–68 (2003, November)
2. Voorhees, E.M.: Evaluating answers to definition questions. In: 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology: companion volume of the Proceedings of HLT-NAACL 2003-short papers, vol. 2, pp. 109–111. (2003, May)
3. David, G.: The AQUAINT corpus of english news text LDC2002T31. Linguistic Data Consortium, Web Download. Philadelphia (2002)
4. Xu, J., Licuanan, A., Weischedel, R.M.: TREC 2003 QA at BBN: answering definitional questions. In: TREC 2003, pp. 98–106 (2003, November)

5. Yang, H., Cui, H., Maslennikov, M., Qiu, L., Kan, M.Y., Chua, T.S.: Qualifier in TREC-12 QA main task. In: TREC 2003, pp. 480–488 (2003, November)
6. Echihabi, A., Hermjakob, U., Hovy, E.H., Marcu, D., Melz, E., Ravichandran, D.: Multiple-engine question answering in TextMap. In: TREC 2003, pp. 772–781 (2003, November)
7. Katz, B., Lin, J.J., Loreto, D., Hildebrandt, W., Bilotti, M.W., Felshin, S., Fernandes, A., Marton, G., Mora, F.: Integrating web-based and corpus-based techniques for question answering. In: TREC 2003, pp. 426–435 (2003, November)
8. Wesley, H., Katz, B., Lin, J.: Answering definition questions with multiple knowledge sources. In: Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics, Boston, Massachusetts, HLT-NAACL (2004)
9. Han, K.S., Song, Y.I., Rim, H.C.: Probabilistic model for definitional question answering. In: 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 212–219. ACM, New York, USA (2006)
10. Cui, H., Kan, M.Y., Chua, T.S.: Generic soft pattern models for definitional question answering. In: 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'05). pp. 384–391, ACM, New York (2005) http://dx.doi.org/10.1145/1076034.1076101
11. Cui, H., Kan, M.Y., Chua, T.S.: Soft pattern matching models for definitional question answering. ACM Trans. Inf. Syst. **25**(2), 8 (2007). https://doi.org/10.1145/1229179.1229182
12. Chen, Y., Zhou, M., Wang, S.: Reranking answers for definitional QA using language modeling. In: 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics, pp. 1081–1088, Sidney (2006)
13. Pasca, M.: Answering definition questions via temporally-anchored text snip-pets. Third Int. Joint Conf. Nat. Lang. Process. **1**, 411–417 (2008)
14. Lita L.V., Hunt W.A., Nyberg E.: Resource analysis for question answering. In: ACL 2004 Interactive Poster and Demonstration Sessions. Association for Computational Linguistics (2004)
15. Macdonald, C., McCreadie, R., Santos, R.L, Ounis, I.: From puppy to maturity: experiences in developing terrier. In: OSIR at SIGIR, pp. 60–63 (2002)
16. Angeli, G., Premkumar, M.J., Manning, C.D.: Leveraging linguistic structure for open domain information extraction. In: 53rd Annual Meeting of the Association for Computational Linguistics (2015, July)
17. Klein, D., Manning, C.D.: Accurate unlexicalized parsing. In: 41st Annual Meeting of the Association for Computational Linguistics (2003)