Answering Factoid Questions in the Biomedical Domain

Dirk Weissenborn, George Tsatsaronis, and Michael Schroeder

Biotechnology Center, Technische Universität Dresden
{dirk.weissenborn,george.tsatsaronis,ms}@biotec.tu-dresden.de

Abstract. In this work we present a novel approach towards the extraction of factoid answers to biomedical questions. The approach is based on the combination of structured (ontological) and unstructured (textual) knowledge sources, which enables the system to extract factoid answer candidates out of a predefined set of documents that are related to the input questions. The candidates are scored by applying a variety of scoring schemes and are combined to find the best extracted candidate answer. The suggested approach was submitted in the framework of the *BioASQ* challenge[1] as the baseline system to address the automated answering of factoid questions, in the framework of challenge *1b*. Preliminary evaluation in the factoid questions of the dry-run set of the competition shows promising results, with a reported average accuracy of 54.66%.

1 Introduction

The task of automatically answering natural language questions (*QA*) dates back to the *1960*s, but has only become a major research field within the information retrieval and extraction (IR/IE) community in the past fifteen years with the introduction of the *QA* Track in *TREC*[2] evaluations in *1999* [1]. The *TREC QA* challenge had its focus on factoid, list and definitional questions, which were not restricted to any domain. A lot of effort has been put into answering the special case of factoid questions, especially by *IBM*, while developing *Watson* [3], a system which was able to compete with, and eventually win, the two highest ranked human players in the famous quiz show *Jeopardy*™[3].

The *BioASQ* challenge [5] differs from the aforementioned challenge in two main points. First, it includes two new types of questions, namely yes/no questions and questions expecting a summary as answer, such as explanations. Second, the questions of the challenge are restricted to the biomedical domain. The restriction to one domain (*RDQA*) induces automatically a specific terminology resulting in questions that tend to be rather technical and complex, but at the same time less ambiguous. In this context, Athenikos and Han [1] report the following characteristics for *RDQA* in the biomedical domain: (1) large-sized textual corpora, e.g., *MEDLINE*, (2) highly complex domain-specific terminology, that is covered by domain-specific lexical, terminological, and ontological resources, e.g., *MeSH*, *UMLS*, and, (3) tools and methods for exploiting

[1] http://bioasq.org/
[2] http://trec.nist.gov/
[3] http://www.jeopardy.com/

the semantic information that can be extracted from the aforementioned resources e.g., *MetaMap*.

Under this scope, in this paper we present a *QA* system that can address with high accuracy the factoid questions in the biomedical domain. The system performs two sequential processes; *Question Analysis*, and *Answer Extraction* which are explained in detail in Section 2. It uses all of the *MEDLINE* indexed documents as a textual corpus, the *UMLS* metathesaurus as the ontological resource, and *ClearNLP*[4], *OpenNLP*[5], and, *MetaMap* as additional tools to process the corpus. The novelty of the work lies in the combined application of several different scoring schemes to assess the extracted candidate answers. In the same direction with the current work, the use of *(weighted) prominence scoring* and *IDF scoring* has been widely applied in the past [6]. The current work is based on the methodology introduced by the IBM's Watson *QA* system, where one of the main characteristics of the system is the combination of a large number of scoring measures, which are combined to extract the most appropriate answer. One prominent example of such a scoring scheme is *type coercion*, which is also utilized in our approach. Towards the direction of combining scoring schemes, we utilize *logistic regression*, which learns the weights for the individual scores. One of the most related works in the same direction is the work by Demner-Fushman and Lin [2], who also utilize *logistic regression* to assess and generate textual responses to clinical questions. However, the main difference with that work lies in the output of the *QA* system, which in our case is a list of biomedical concepts provided as answers to factoid questions, while in the work presented in [2] the output is textual responses, thus differentiating substantially the core of the answer extraction methodology and ranking.

2 A *QA* system for factoid questions in the biomedical domain

Initially, a *question analysis module* analyses the question, by identifying the *lexical answer type* (*LAT*), given a factoid or list question. The *LAT* is the type of answer(s) the system should search for. Next, documents relevant to the question are retrieved by the *document or passage retrieval module*, which takes the input question and transforms it into a keyword query. For this step, we rely on the results provided by the *GoPubMed* search engine[6]. In a last step, the top N retrieved documents are being processed by the *answer extraction module*. This module finds possible answer candidates within the relevant documents and scores them. These scores are combined to a final score which serves as the basis for ranking all answer candidates.

2.1 Question Analysis

Question Analysis is responsible for extracting the *LAT* (lexical answer type), which is crucial for understanding what the question is actually asking about. In the English language questions follow specific patterns, thus making it easy to apply pattern based

[4] `https://code.google.com/p/clearnlp/`
[5] `http://opennlp.apache.org/`
[6] `http://www.gopubmed.org/web/gopubmed/`

extraction methods. However, there are different kinds of factoid questions, like the following examples illustrate:

Example 1. What is the methyl donor of DNA (cytosine-5)-methyltransferases?

Example 2. Where in the cell do we find the protein Cep135?

Example 3. Is rheumatoid arthritis more common in men or women?

From this perspective, the questions answerable by the current system fall into one of the three following classes:(1) *what/which-questions* (Example 1), (2) *where-questions* (Example 2), and, (3) *decision-questions*, (Example 3). Factoid or list questions usually carry information about the type of the answer (the *LAT*). The phrase containing the *LAT* is found directly after the question word (what/which), or after the question word followed by a form of "be". *Where-questions* do not have an explicitly defined *LAT*; instead, they implicitly expect spatial concepts as answers. There are also cases in which *where-questions* restrict the set of potential answer types further, by defining a place where the (spatial) answer concept should be part of, e.g. "the cell" in Example 2. *Decision-questions* also do not have a *LAT*. It rather consists of a number of possible answers, from which one has to be selected, e.g., "men", "women". Furthermore they define a criterion by which the answer candidates have to be differentiated, e.g., "more common". The extraction rules for all types of questions are based on the chunks and the dependency parse of the question. For instance Example 1 falls into the following pattern: *'NP[What|Which] VP[BE] NP[*] *?'*, where * serves as a wildcard, *NP* stands for nounal phrase and *VP* for verbal phrase. The *LAT* of questions falling into this pattern can be found in the second *NP*.

2.2 Answer Extraction

After the relevant documents for the question have been retrieved, all annotated concepts of these documents are considered possible answer candidates. To extract the right answer the candidates are scored and ranked. The proposed system supports 6 different scores of 3 different types (*prominence, type, specificity/idf*). In the following, the different scores are explained in detail. The following notation is utilized: q is the input question, *ac* is a candidate answer, *D* is the document set that is relevant to the question, *S* is the set of all sentences in *D*, and *A* is the set of all documents present in the used corpus (*MEDLINE*).

Prominence Scoring *Prominence Scoring* is based on the hypothesis that the answer to a question appears often in the relevant documents. Thus, this score counts all occurrences of a concept within the sentences of all of the relevant documents, and can be formalized as follows:

$$score_{pr}(ac) = \frac{\sum_{s \in S} \mathbb{I}(ac \in s)}{|S|}, \tag{1}$$

where $\mathbb{I}$ is the indicator function, returning 1 if the argument is true and 0 otherwise. The problem with this score is the assumption that each sentence is equally relevant to the question. This is a very poor assumption giving rise to the refined hypothesis that the answer to a question appears often in the relevant sentences. The refined hypothesis can be formulated by weighting sentences according to their similarity to the question, as shown in the following equation:

$$score_{wpr}(ac) = \frac{\sum_{s \in S} sim(q,s) \cdot \mathbb{I}(ac \in s)}{\sum_{s \in S} sim(q,s)} \tag{2}$$

The *weighted prominence score (wpr)* makes use of a similarity measure for sentences. The way it is implemented in the proposed system is by measuring the number of common concepts between the question and a sentence, normalized by the number of question concepts. This measure has also been used in other *QA* systems [6].

Specificity Scoring Prominence scores boost very common concepts like *DNA* or *RNA*, which appear quite often. Thus, the *specificity score* is introduced as an additional scoring mechanism, based on the hypothesis that the answer to a biomedical question appears in a rather small number of documents. The *specificity score* is a simple *idf-based (inverted document frequency) score* on the whole *MEDLINE* corpus of abstracts normalized by the maximum *idf* score. The formula is shown in the following equation:

$$score_{idf} = log(\frac{|A|}{\sum_{a \in A} \mathbb{I}(ac \in a)})/log(|A|) \tag{3}$$

Type Coercion *Type Coercion* is based on the hypothesis that the type of the answer aligns with the lexical answer type (*LAT*). *Type coercion* was introduced by *IBM* within the *Watson* system [4]. Rather than restricting the set of possible answers beforehand to all candidates that are instances of the *LAT*, *type coercion* is used as a scoring component which tries to map the answer candidates to the desired type employing a variety of strategies. In the following, we analyze the two strategies used to implement the *type coercion* in the proposed system, namely: *UMLS Type Scoring*, and *Textual Type Scoring*.
The *UMLS type score* is based on the *UMLS* semantic network. Each *UMLS* concept of the metathesaurus has a set of semantic types assigned to it. These semantic types are hierarchically structured in the semantic network. If the question analysis module finds the *LAT* to be a *UMLS* concept, the semantic types of this concept and its children in the hierarchy of the semantic network become the target types of the question. If the candidate answer's semantic types and the target types have common types, the *UMLS type score* will be 1, and 0 otherwise. This is shown analytically in the following equation:

$$score_{umls} = \begin{cases} 1 & \text{if } sem\_types(ac) \cup sem\_types(LAT) \neq \oslash \\ 0 & else \end{cases} \tag{4}$$

The *textual type scorer* tries to find 3 types of connections between the *LAT* and the answer candidate. The most obvious pattern is the one where the answer candidate is a subject and the the *LAT* is an object, connected via a form of “be”. Two other syntactic constructions, namely *nominal modifiers* and *appositions*, are used to infer the type of the answer candidate. The following examples illustrate the 3 cases:

– ... naloxone is standard medication... [BE]
– ... the medication naloxone... [NOMINAL MOD.]
– ... naloxone, a medication... [APPOS]

All of the above examples are evidence for *naloxone* being a medication. However, *nominal modifiers* and *appositions* are not that reliable, so the *textual type score* for these cases should be lower. The scoring is described analytically in the following:

$$score_{pa} = \begin{cases} 1 & \text{if } "ac - BE - type" \in D \\ 0.5 & \text{if } "ac - appos\ or\ nom.mod. - type" \in D \\ 0 & else \end{cases} \tag{5}$$

If there is no textual evidence in the retrieved documents D, there could still be this kind of evidence in some other *MEDLINE* abstracts (A). This kind of evidence is called *supporting evidence*, because it searches also in unrelated documents for evidence. The following equation shows how the *supporting evidence* is scored:

$$score_{supp} = \begin{cases} 1 & \text{if } "ac - BE - type" \in A \\ 0.5 & \text{if } "ac - appos\ or\ nom.mod. - type" \in A \\ 0 & else \end{cases} \tag{6}$$

Eventually, in the proposed system the top 5 retrieved supporting documents for each answer candidate are examined.

Score Merging After scoring the answer candidates, the scores have to be combined into a final score. In the proposed system, we followed a supervised approach to learn appropriate weights for the individual scores. Given a dataset of factoid questions and their "gold" answers, a training set consisting of positive (right answers) and negative (wrong answers) examples represented by their scores is constructed in order to train a logistic regression classifier. Hence, the training instances are answers, and the features are the scores of the individual measures. The logistic regression learns appropriate weights for each of the represented features (scores). The learned weights of the output classifier (learned model) are then used as the weights for each of the scoring measures respectively. The final score is produced by applying the learned logistic regression formula.

Candidate Selection and Answer Suggestion For factoid questions the candidate with the highest final score is chosen to be the answer. However, in the case of list questions it is not easy to define the right cut-off in the list manually, e.g., how many items to suggest as answer starting from the beginning of the list. For this purpose, we employed again a supervised approach, where the threshold on the final score is estimated by calculating the F_1 score on all training list-questions for all possible thresholds within a certain interval. The threshold with the highest F_1 score is finally chosen to be the cut-off for list questions.

3 Experimental Evaluation and Preliminary Results

Experiments were conducted on the dry run test set of the *BioASQ Challenge 1b* [5]. It comprises 29 questions of which 12 were factoid or list questions. Only 8 out of these 12 questions had answers which are part of the *UMLS* metathesaurus, thus they were the only actually answerable questions for the proposed system. Following a fold-cross validation approach for the training, the system obtained an overall average accuracy of 54.66% in these 12 questions. Isolating the evaluation only on the 8 questions the system was able to answer, an average accuracy of 82% was achieved.

4 Conclusions

The results of the experiments, though preliminary in the sense that they were conducted in the dry-run set of the *BioASQ* challenge, indicate that the suggested baseline approach is able to achieve reasonable performance on answerable factoid questions. However, there is still a lot of room for improvements, like considering more scores and making the system faster to handle all supporting evidence. More sophisticated systems for factoid question answering, similar to the *IBM Watson* system, but especially designed for the biomedical domain, can be developed given the huge amount of structured and unstructured resources in the domain, but this requires large infrastructure, since *QA* pertains to nearly every aspect of information extraction and natural language processing. In addition, the effect of adding larger number of training questions will be investigated in the future, once the benchmark questions from the first *BioASQ* challenge are released to the public. The proposed system was submitted as a baseline for challenge *1b* of the *BioASQ* competition to address exclusively factoid questions, and as a future work, we plan to analyze the results on the actual test set of the competition and study how further improvements may be achieved in an effort to address more effectively factoid questions in the biomedical domain.

References

1. S. J. Athenikos and H. Han. Biomedical question answering: a survey. *Computer methods and programs in biomedicine*, 99(1):1–24, Jul 2010.
2. D. Demner-Fushman and J. Lin. Answering clinical questions with knowledge-based and statistical techniques. *Comput. Linguist.*, 33(1):63–103, Mar. 2007.
3. D. Ferrucci. Introduction to ”This is Watson”. *IBM Journal of Research and Development*, 56(3.4):1:1–1:15, 2012.
4. J. W. Murdock, A. Kalyanpur, C. Welty, J. Fan, D. A. Ferrucci, D. C. Gondek, L. Zhang, and H. Kanayama. Typing candidate answers using type coercion. *IBM J. Res. Dev.*, 56(3):312–324, May 2012.
5. G. Tsatsaronis, M. Schroeder, G. Paliouras, Y. Almirantis, I. Androutsopoulos, E. Gaussier, P. Gallinari, T. Artieres, M. R. Alvers, M. Zschunke, et al. Bioasq: A challenge on large-scale biomedical semantic indexing and question answering. In *2012 AAAI Fall Symposium Series*, 2012.
6. M. Wang. A survey of answer extraction techniques in factoid question answering. In *Proceedings of the Human Language Technology Conference and North American Chapter of the Association for Computational Linguistics (HLT-NAACL)*, 2006.