Type Checking in Open-Domain Question Answering (Extended Abstract)

Stefan Schlobach   Marius Olsthoorn   Maarten de Rijke

Informatics Institute, University of Amsterdam
schlobac,olstrn,mdr@science.uva.nl

Abstract

The full version of this paper appeared in *Proceedings ECAI 2004*, Valencia, 2004. IOS Press.

Question answering (QA) is one of several recent attempts to realize information *pinpointing* as a refinement of the traditional document retrieval task. In response to a user's question, a QA system has to return an answer instead of a ranked list of relevant documents from which the user has to extract an answer herself.

Ontology-based Answer Type-Checking. Open domain QA systems have to bridge the potential vocabulary mismatch between a question and its candidate answers. One can view this as a *recall* problem and address it accordingly. Recall oriented strategies to QA may generate considerable amounts of noise. To combat this, many open domain QA systems contain a filtering or re-ranking component, and in many cases this involves checking whether the answer is of the correct semantic type. Particular classes of questions expect specific answer types to which all of their answers should belong. The *expected answer type(s)* (or EAT(s)) of a question restrict(s) the admissible answers within a particular domain, such as the geography domain, to more specific classes, such as *river* or *country.* In our approach, answer types are WordNet synsets. The EATs of a question can often be reliably determined by simple extraction patterns. We compare two strategies for answer type checking. One is redundancy-based reranking and uses the redundancy of information available on the web to estimate the amount of implicit knowledge which connects an answer to a question. The other is knowledge-intensive filtering and exploits structured and semi-structured data sources to determine the semantic type of suggested answers.

Filtering and Re-Ranking. If a candidate answer is known *not* to be an instance of any EAT associated with a question, it can immediately be excluded from the answer selection process. We will refer to this use of EATs as *answer type checking* by *filtering.* For filtering, a knowledge-intensive approach seems ideally suited: for each candidate answer we try to extract a *found answer type (FAT)* from knowledge and data sources, i.e., a most specific semantic type of which it is an instance. To determine the FATs of an answer we use WordNet and two Geographical Name Servers (GNS and GNIS) as external data sources. An answer is kept if there is a FAT that is at least as specific as one of the EATs.

Because of the inherent incompleteness of knowledge and data sources in open domain applications, it may be impossible to determine a FAT for every candidate answer. Instead, we propose to determine the likelihood that the expected answer type is indeed a correct semantic type for a candidate answer and to *re-rank* the candidate answers according to this measure. For re-ranking, redundancy-based strategies are an obvious choice, the assumption being that the number of co-occurrences of answers and answer types allows us to quantify the relation between a question's EAT and a candidate answer. As statistical measures we use *conditional type probability*: $CTP(E|A) = P(E, A) = \frac{hc(E+A)}{hc(A)}$, where $hc(T)$ is the *hit-count* of T, i.e., the number of web pages on which a term T occurs. That is, the probability that the expected answer type E occurs in a document given that it contains the candidate answer A. Secondly, we introduce *normalized conditional type probability (NCTP)* which in addition normalizes over the occurrences of hits of the answer.

Experiments. We evaluated the output of our QA system on 839 location questions; the list of candidate answers returned by the system was subjected to answer type checking. To establish an upper-bound on the performance of answer type checking we determined how much *human type-checking* can improve the results. Then, we compared the performance of knowledge intensive filtering and redundancy based re-ranking. To study the influence of the use of databases on filtering, we ran a dressed down version of algorithm to find the FATs, using only WORDNET. We denote the latter method by KIF-WN, and the full version as KIF. The experimental results show that type checking KIF can significantly improve the overall performance of a QA system for geography questions, but that even the best available strategy performs significantly worse than a human expert. Redundancy based re-ranking failed to make a difference on the overall performance. Both problems can be explained by the semantic ambiguity of the candidate answer; the types of candidate answers are determined incorrectly, essentially because no use is being made of the question's context. We have explored two methods combining knowledge intensive and redundancy-based approaches and implemented one of them with promising first results. Our evaluation is specific for the geography questions that we considered — this is an ideal domain for knowledge intensive approaches. To port our approaches to other domains an ontology of types, mechanisms to extract the EATs and mappings to FATs are required. The redundancy-based approach is obviously domain independent. Hence, we expect to be able to apply substantial parts of our general strategy in other domains.

Strategy	correct answers	% correct answers
No type-checking	244	29%
Human type-checking	331 (+36%)	36.4%
KIF	271 (+11%)	32.3%
KIF-WN	292 (+20%)	34.8%
RBRR-CTP	248 (+2%)	30%
RBRR-NCTP	249 (+2%)	30%

Acknowledgments. Thanks to Gilad Mishne for help. This research was supported by the Netherlands Organization for Scientific Research (NWO) under project number 220-80-001. Maarten de Rijke was also supported by NWO under project numbers 612-13-001, 365-20-005, 612.069.006, 612.000.106, 612.000.207, 612.066.302, and 264-70-050.