Answer Finding Guided by Question Semantic Constraints

Marius Paşca
Language Computer Corporation
Dallas, Texas
marius@languagecomputer.com

Abstract

As part of the task of automated question answering from a large collection of text documents, the reduction of the search space to a smaller set of document passages that are actually searched for answers constitutes a difficult but rewarding research issue. We propose a set of precision-enhancing filters for passage retrieval based on semantic constraints detected in the submitted questions. The approach improves the performance of the underlying question answering system in terms of both answer accuracy and time performance.

Introduction

The introduction of the Question Answering track in the Text REtrieval Conference (TREC) (Voorhees 1999) provided a big boost to research in open-domain Question Answering (QA). A QA system accepts natural language questions as input, thus eliminating the need to understand a particular query language or artificial syntax (e.g., Boolean operators) before querying the underlying text collection. In response to the user's questions, the QA system returns a set of short answers, capturing directly the piece of information that actually matches the user's information need. This is an important advantage over document retrieval systems, which return long lists of documents even though the information that is actually relevant is often concentrated within small document fragments.

The extraction of answer strings from large text collections consists of three main sub-tasks. *Question processing* deals with the transformation of the raw question string into a higher-level representation that is more useful to answer finding. *Passage retrieval* reduces the search space. Rather than applying expensive answer extraction procedures to the entire document collection - an unfeasible solution with Gigabyte collections - passage retrieval limits the search to a much smaller set of document passages where answers may exist. *Answer extraction* consists in the exploration of the document content in order to identify, extract and rank the relevant answer strings that are returned to the users. Many recent systems feature modules that are dedicated to question processing, passage retrieval and answer extraction (Abney, Collins, & Singhal 2000; Hovy *et al.* 2001; Moldovan *et al.* 2000).

A middle layer in the QA system and usually invisible to the user, the passage retrieval module affects the overall performance both qualitatively (precision) and quantitatively (execution time). The number of passages retrieved from the collection should be as small as possible such that a small quantity of text will be mined in search for the actual answers. Conversely, passage retrieval should not be too restrictive, or passages containing correct answers may be missed.

Our experience in the TREC QA track has shown that the mere occurrence of the lexical terms used in the question, within the same document passage, does not guarantee that the passage actually contains an answer. Furthermore, if the answers are searched in very large collections (e.g., the World Wide Web), then thousands of different documents may all contain the same keywords, even though only a few actually contain relevant answers. A necessary condition for the passages to be relevant is that they also satisfy the constraints that users often specify implicitly in their questions.

In this paper, we propose a set of precision-enhancing filters for passage retrieval based on semantic constraints automatically detected in the submitted questions. While the passage filtering does not guarantee the absolute relevance of the passages, it does filter out many spurious text fragments and thus improves both the time performance and the quality of the output. The approach is evaluated on a set of 893 questions whose answers are extracted from a 3-Gigabyte text collection.

Question Semantic Constraints

Figure 1 captures the impact of the semantic constraints detected in the question, on the information items that are passed internally through the system. The submitted question, *"How much could you rent a Volkswagen bug for in 1966?"*, is parsed and then transformed into the question semantic representation (Paşca 2001) containing question terms connected to each other through binary relations. In particular, the relation between the question stem *How much* and the question term *rent* allows for the derivation of the expected answer type or the category of the expected answers, namely MONEY. The question keywords are used for the construction of the Boolean query "Volkswagen AND bug", which is passed to the passage retrieval engine. The engine's output consists of sixty text passages retrieved from

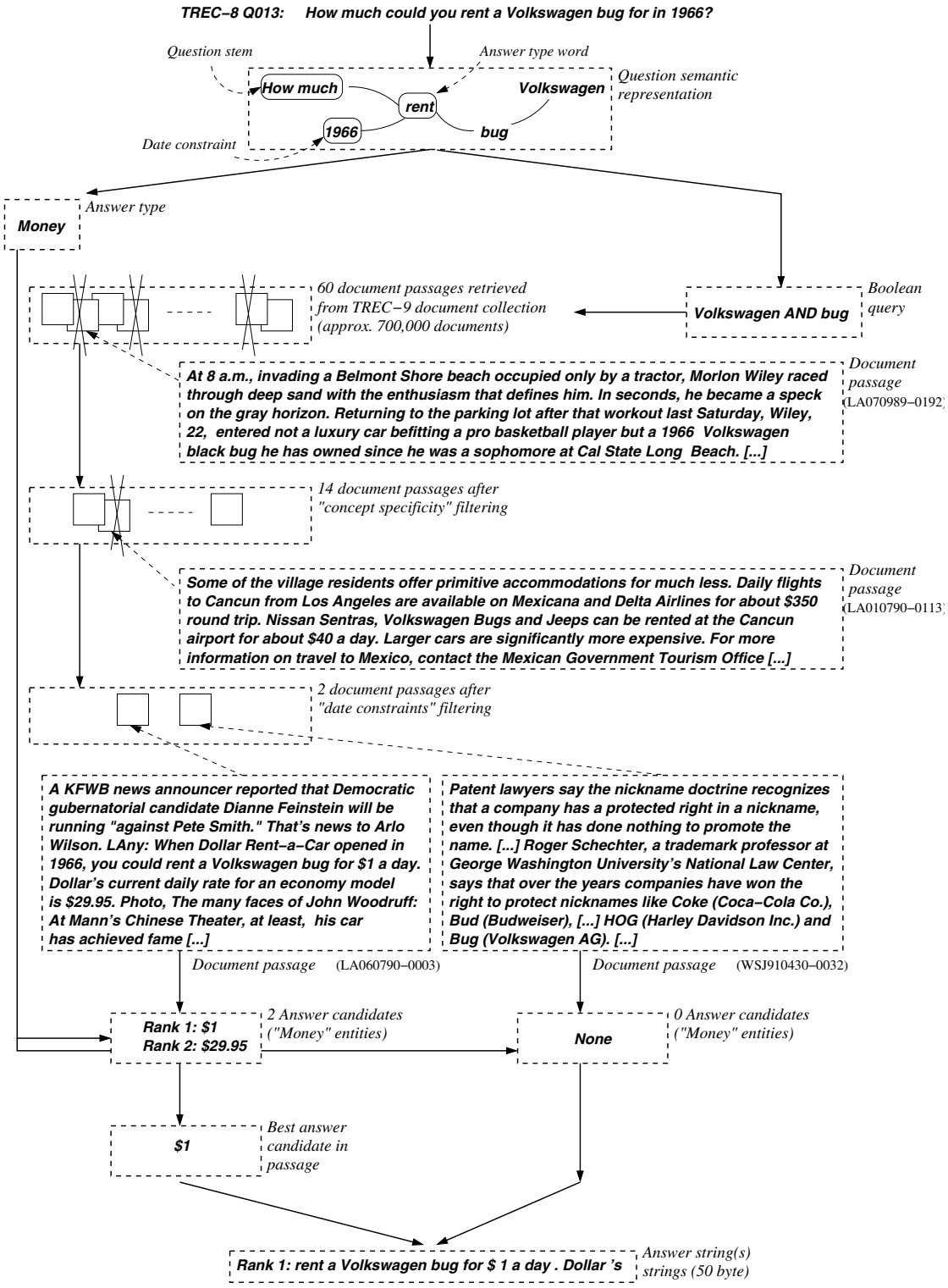

Figure 1: Exploiting question semantic constraints: answering TREC-8 test question Q013

the TREC-9 document collection.

Because the question term *rent* is deemed to be very specific, all passages which do not contain the term are rejected after the application of the concept specificity filter. Note that one of the rejected passages, from the Los Angeles Times article LA070989-0192, mentions a basketball player who *owns* a Volkswagen bug rather than *rents* it (see Figure 1). Similarly, the absence of the date constraint *1966* from any of the text passages causes the elimination of the passage. This turns out to be very helpful in rejecting passages such as that from article LA010790-0113, which does refer to the rental price of Volkswagen bugs, but not the price from 1966. As a combined effect of the passage semantic filters, the initial set of sixty passages is reduced to only two passages, both of which are shown in the figure.

The first of the re-ranked, filtered document passages contains two entities of the expected answer type MONEY, i.e., *$1* and *$29.95*. The returned answer strings are extracted from the text passages around the highest ranking candidate answer.

Experimental Setting

The experiments involve the standard test collection from the TREC QA track. The question set consists of fact-seeking, short-answer questions from the TREC QA track. There are 200 questions from TREC-8 and 693 questions from TREC-9, all of which are guaranteed to have at least one answer in the underlying document collection. The document collection comprises more than 700,000 text documents from sources including Los Angeles Times and Wall Street Journal.

Individual questions are assigned a score equal to the reciprocal answer rank. The answer rank is the rank of first correct 50-byte answer returned by the system (Voorhees 1999). Thus a question receives a score of 1, 0.5, 0.33, 0.25, 0.2, if the first correct answer is returned at rank 1, 2, 3, 4 and 5 respectively. By convention, the score for questions with no correct answer among the first five returned is 0.

The overall accuracy or *precision score* is the mean of the reciprocal answer ranks computed over the entire set of questions. To ensure that the results are relevant to state-of-the-art QA, we use in our experiments the same QA technology that answered correctly the highest number of questions in the TREC-8 and TREC-9 QA track evaluations.

Conceptual Filtering

Terms such as *rent* from Q013: *"How much could you rent a Volkswagen bug for in 1966?"*, or *wage* from Q296: *"What is the federal minimum wage?"* play a special role in answer finding. The semantic disambiguation of these terms, formally called answer type terms, into their corresponding answer type (MONEY) relies on the mapping of the question semantic representations onto a lexico-semantic hierarchy of answer types. The passages without any occurrence of a concept of the expected answer type are discarded as irrelevant. Furthermore, the concepts of the expected answer type that are also in the proximity of the question terms are likely to constitute an answer and scored accordingly. When the semantic disambiguation of answer type terms is enabled, the overall precision of the QA system on the TREC-8 and TREC-9 questions combined improves substantially as shown in Table 1.

Table 1: Impact of conceptual filtering based on semantically disambiguated answer type terms on QA precision

Filter usage	Precision score (893 questions)
Disabled	0.487
Enabled	0.565

The accuracy of the extracted answers can be further enhanced based on concept specificity. The more specific the answer type terms from the questions are, the more likely it is for them to occur in similar forms in the relevant answer strings. For Q056: *"How many calories are there in a Big Mac?"*, which makes reference to the very specific answer type term *calories*, the answer contains the answer type term: *"A Big Mac has* 562 *calories"*. Similarly, in the case of the question Q013: *"How much could you rent a Volkswagen bug for in 1966?"*, the answer contains the term *rent*: *"in 1966, you could rent a Volkswagen bug for $1 a day"*.

The estimation of whether a concept is specific or not could rely on the depth of the concept in the WordNet (Miller 1995) hierarchies, or on the coverage of that concept measured as number of hyponyms - or more specific concepts. After a few experiments, we realized that the depth alone does not capture concept specificity well, due to the lack of balance in WordNet hierarchies. Consequently, we developed an algorithm for specificity computation following the second alternative (Paşca 2001):

Step 1. Identify the question term that determines the answer type: *a-term*
Step 2. Take all senses of *a-term* from WordNet: aw$_i$, i=$\overline{1..n}$
Step 3. For each sense aw_i, i=$\overline{1..n}$
  3.1. Collect all hyponyms of aw_i in a set S
  3.2. Remove from S those hyponyms containing *answer-word* as head word
  3.3. Remove from S those hyponyms that are proper nouns (names)
Step 4. Decide upon specificity of *a-term*
  4.1. If card(S) $< Threshold$, *a-term* is specific
  4.2. If card(S) $\geq Threshold$, *a-term* is not specific

The algorithm deems a concept as very specific if it has few hyponyms in its sub-hierarchy. Note that the algorithm takes into account (1) the mismatch between the word level (for answer type terms) and conceptual level (Step 1); (2) the treatment of compound WordNet concepts during specificity estimation (Step 3.2); and (3) the blurring of the KindOf and InstanceOf relations within the WordNet hierarchies (Step 3.3).

Table 2 illustrates the application of passage post-filtering for three TREC questions with specific answer type terms. After discarding the retrieved passages that do not contain

Table 2: Discarding irrelevant passages through specificity-based passage filtering

Question (answer type term in italics)	#Passages retrieved (Boolean query)	Filter	#Passages retained
Q043: What costume *designer* decided that Michael Jackson should only wear one glove?	(Michael ∧ Jackson ∧ ∧ costume) ⇒ 50	designer	1 (2%)
Q230: When did the vesuvius last *erupt*?	(vesuvius ∧ last) ⇒ 25	erupt	13 (52%)
Q231: Who was the *president* of Vichy France?	(Vichy ∧ France) ⇒ 129	president	51 (39%)

Table 3: Impact of conceptual filtering based on highly specific answer type terms

Question (answer type term in italics)	Precision score: filter disabled	Precision score: filter enabled
Q016: What two US *biochemists* won the Nobel Prize in medicine in 1992?	0.33 (rank: 3)	1 (rank: 1)
Q168: Who was the *captain* of the tanker, Exxon Valdez, involved in the oil spill in Prince William Sound, Alaska, 1989?	0 (rank: not first 5)	1 (rank: 1)
Q800: What *monarch* signed the Magna Carta?	0 (rank: not first 5)	1 (rank: 1)
Q506: Who *reports* the weather on the "Good Morning America" television show?	1 (rank: 1)	0 (rank: not first 5)

the answer type term, we obtain a significant reduction over the number of passages retrieved. For the example from Figure 1, among the 60 passages retrieved by the query (Volkswagen ∧ bug), only 14 contain the answer type term *rent* which is very specific (it has only one hyponym - a more specific concept - in WordNet). The 46 passages that are discarded because they do not contain the specific answer type represent an important reduction ratio (77%) over the initial set of 60 retrieved passages. Table 2 shows other questions for which irrelevant passages are successfully filtered out based on conceptual specificity.

When run on the 893 TREC evaluation questions, the specificity estimation procedure finds very specific answer types for 366 questions. In turn, the 366 questions are split into 258 questions for which the answer type term is already used in the Boolean query for passage retrieval, and 108 questions for which the answer type is not used in the Boolean query. For the latter 108 questions, the average number of passages retrieved in the last iteration is 346. Comparatively, the average number of passages retained after specificity post-filtering is 52, corresponding to a passage rejection / retention rate of 63%/37%. The strongest filtering occurs for Q830: *"What is the equivalent of the Red Cross in the Middle East?"*, for which only 4 out of 409 retrieved passages are retained after post-filtering (99% of the passages are discarded). On the other hand, no passage is rejected for Q469: *"Who coined the term "cyberspace" in his novel "Neuromancer"?"* (one passage is retrieved and also retained). In addition to better time performance after reducing the number of passages, the conceptual filtering based on answer type term specificity also improves the quality of the extracted answers. When specificity-based passage filtering is enabled, the overall precision score increases by 1.7% for TREC-9 questions only, and 1.5% for TREC-8 and TREC-9 questions. The individual precision score changes for 31 questions: it increases for 27, and decreases for 4 questions. Thus there are few cases (4 out of 366) when the application of the filter actually leads to the extraction of irrelevant answer strings due to the rejection of all relevant passages. Table 3 illustrates some of the 31 questions whose score is affected.

Date Constraints

In addition to specific answer type terms, another type of semantic constraint that is useful for passage post-filtering are the date constraints, e.g., *"How much could you rent a Volkswagen bug for in 1966?"*. The detection of dates in the submitted questions is performed with a named entity recognizer which supports the *date* and year entity types. When a date constraint is found in the question, the retrieved passages are analyzed with the following algorithm:

Step 1. Identify the date constraints from the question: *yearQuestion*
Step 2. If there is not exactly one date *yearQuestion*, then retain all passages and return
Step 3. For each passage P_i retrieved from the collection, i=$\overline{1..n}$
  3.1. If *yearQuestion* occurs as a token in P_i, then retain P_i
  3.2. Else if *minCollectionYear* $\leq$ *yearQuestion* $\leq$ $\leq$ *maxCollectionYear*
    3.2.1. Extract creation date for document containing P_i: *yearDoc*
    3.2.2. If *yearDoc* $<$ *yearQuestion* then discard P_i

3.2.3. Else retain P_i (defensive approach)
3.3. Else if *yearQuestion* $\leq$ *minCollectionYear* then discard P_i
3.4. Else discard P_i.

The algorithm correlates the date stamp of each of the retrieved passages with the date constraint specified in the question. In the case of TREC documents, this information appears in the document header. The earliest and the latest published documents determine the time range of the collection, denoted by *minCollectionYear* and *maxCollectionYear*. Each passage is retained or discarded, depending on the relation between the date of the passage and the date constraint from the question.

Table 4: Passage filtering based on date constraints

Question	#Passages retrieved/ discarded/ retained (pct. retained)
Q036: In *1990*, what day of the week did Christmas fall on ?	496/393/103 (21%)
Q040: Who won the Nobel Peace Prize in *1991*?	1133/991/142 (13%)
Q063: What nuclear-powered Russian submarine sank in the Norwegian Sea on *April 7, 1989*?	23/16/7 (30%)
Q103: How many people died when the Estonia sank in *1994*?	259/201/58 (22%)

As shown in Table 4, verifying the date constraints is another source of semantic information that is very useful for refining the output from the passage retrieval engine. When run on the 200 evaluation questions from the TREC-8 QA track, the algorithm identifies 25 questions containing date constraints. The precision score of the system changes for 5 out of the 25 TREC-8 questions, when verification of date constraints is enabled. The precision scores change as follows: 0.2 to 0.5 for Q004, 0.33 to 1 for Q016, 0.2 to 0.5 for Q036, 0.5 to 1 for Q040, and 0 to 0.25 for Q169. From a total of 6017 passages retrieved for the 25 questions, 1220 passages are retained after post-filtering based on date constraints. This corresponds to an average of 80% passages being discarded, per question specifying a date contraint. Consequently, the answer extraction module has to process fewer passages and runs 45% faster. The impact of date constraints is negligible on the TREC-9 questions, because only 4 out of the 693 evaluation questions specify a date constraint, e.g., Q242: *"What was the name of the famous battle in 1836 between Texas and Mexico?"*.

Conclusions

This paper presented a method for enhancing the precision of a large-scale QA system based on advanced processing of natural language questions. The method exploits the information extracted automatically from the submitted questions in order to filter the retrieved passages, before they are actually searched for answers. Previous approaches acknowledge the usefulness of the question's answer type in answer extraction (Abney, Collins, & Singhal 2000; Hovy *et al.* 2001). However, to our knowledge there is no recent work focusing on semantic refinements of passage retrieval for large-scale, open-domain QA.

Our approach is specifically designed for large-scale QA systems. Nevertheless, it can be easily extended to any precision-oriented information retrieval system that accepts natural language input, even if the output is not a set of answers but larger text excerpts.

References

Abney, S.; Collins, M.; and Singhal, A. 2000. Answer extraction. In *Proceedings of the 6th Applied Natural Language Processing Conference (ANLP-2000)*, 296–301.

Hovy, E.; Gerber, L.; Hermjakob, U.; Lin, C.; and Ravichandran, D. 2001. Toward semantics-based answer pinpointing. In *Proceedings of the Human Language Technology Conference (HLT-2001)*.

Miller, G. 1995. WordNet: a lexical database. *Communications of the ACM* 38(11):39–41.

Moldovan, D.; Harabagiu, S.; Paşca, M.; Mihalcea, R.; Gîrju, R.; Goodrum, R.; and Rus, V. 2000. The structure and performance of an open-domain question answering system. In *Proceedings of the 38th Annual Meeting of the Association of Computational Linguistics (ACL-2000)*.

Paşca, M. 2001. *High-Performance, Open-Domain Question Answering from Large Text Collections*. Ph.D. Dissertation, Southern Methodist University, Dallas, Texas.

Voorhees, E. 1999. The TREC-8 Question Answering track report. In *Proceedings of the 8th Text REtrieval Conference (TREC-8)*, 77–82. Gaithersburg, Maryland: NIST.