Answer Formulation for Question-Answering

Leila Kosseim[1], Luc Plamondon[2] and Louis-Julien Guillemette[1]

[1]Concordia University
1455 de Maisonneuve Blvd. West, Montréal (Québec) Canada, H3G 1M8
{kosseim, guillem}@cs.concordia.ca
[2]RALI, DIRO, Université de Montréal
CP 6128, Succ. Centre-Ville, Montréal (Québec) Canada, H3C 3J7
plamondl@iro.umontreal.ca

Abstract In this paper, we describe our experimentations in answer formulation for question-answering (QA) systems. In the context of QA, answer formulation can serve two purposes: improving answer extraction or improving human-computer interaction (HCI). Each purpose has different precision/recall requirements. We present our experiments for both purposes and argue that formulations of better linguistic-quality are beneficial for both answer extraction and HCI.

1 Introduction

Recent developments in open-domain question answering (QA) have made it possible for users to ask a fact-based question in natural language (eg. *Who was the Prime Minister of Canada in 1873?*) and receive a specific answer (eg. *Alexander Mackenzie*) rather than an entire document where they must further search for the specific answer themselves. In this respect, QA can be seen as the next generation of daily tools to search huge text collections such as the Internet.

To date, most work in QA has been involved in *answer extraction*; that is, locating the answer in a text collection. In contrast, the problem of *answer formulation* has not received much attention. Investigating answer formulation is important for two main purposes: human-computer interaction (HCI) and answer extraction. First, answer formulation can improve the interaction between QA systems and end-users. As QA systems tackle more difficult issues and are extended to dialog processing systems, a text snippet or a short answer will not be enough to communicate naturally with the user; a full natural sentence that is linguistically motivated will be required. On the other hand, answer formulation can be used as a reverse engineering method to actually improve answer extraction from a large document collection. For example, when looking for the answer to *Who was the Prime Minister of Canada in 1873?* and knowing that the answer could have the form `"In 1873, the Prime Minister of Canada was` <PERSON-NAME>`"` or `"In 1873,` <PERSON-NAME> `was the Prime Minister of Canada"`, the QA system can search for these formulations in the document collection and instantiate <PERSON-NAME> with the matching noun phrase.

Depending on the purpose of answer formulation, different goals will be enhanced at the expense of others. Answer formulations used to extract answers will need to have a high recall rate. The goal here is to produce a large number of possible formulations hoping that one of them will retrieve an answer. If the system produces formulations that are linguistically incorrect or awkward, the consequences are not great; the information retrieval component will simply not find any occurrence of the answer pattern. On the other hand, answer formulation performed to improve HCI will need to aim for high precision. The goal here is not to produce a great number of approximate formulations, but only a few (or only one) of good linguistic quality.

2 Previous Work in Answer Formulation

The field of QA has been chiefly driven by the DARPA initiative through the Text Retrieval Conferences (TREC) [VH99,NIS00,VH01,VH02]. This is why most work has been concentrated on issues related to question parsing (what are we looking for?), information retrieval (what document contains the answer?), and answer extraction (what is the specific answer?). Some research teams follow a more knowledge-rich approach ([HMP+01,HHL01]), while others use statistical approaches ([LCC01]). However, regardless of how the steps are performed, the goal is always to extract an answer or a text snippet, rather than to compose an answer.

The need to investigate answer formulation has already been felt by the QA community [ea01]; however, to our knowledge, little research has yet addressed this issue. A first step toward answer formulation was done at the previous TREC-10 conference, where several teams saw the value of the web as a tremendous source of additional texts to improve answer extraction (eg. [CCL+01,BLB+01]) and as part of work in query expansion to improve information retrieval [AG00,LG98]. In the work of [BLB+01,BDB02], the system searches the web for a list of possible answer formulations generated by permuting the words of the questions. For example, given a question of the form:

$$\textit{Who is } w_1\ w_2\ w_3\ \ldots w_n\textit{?}$$

the system will generate:

$$\begin{array}{c} \texttt{"}w_1\ \texttt{is}\ w_2\ w_3\ \ldots w_n\texttt{"} \\ \texttt{"}w_1\ w_2\ \texttt{is}\ w_3\ \ldots w_n\texttt{"} \\ \texttt{"}w_1\ w_2\ w_3\ \texttt{is}\ \ldots w_n\texttt{"} \\ \ldots \end{array}$$

and will search the web for such phrases. Given the question: `"Who is the world's richest man married to?"`, the following phrases will be searched for: `""the is world's richest man married to""`, `""the world's is richest man married to""`, `""the world's richest man is married to""`... Hopefully, at least one phrase (more likely, the last one in our example) will retrieve

the expected answer. Although simple, this strategy is very efficient. Using this method, [BLB+01] received the 9th best score out of 37 teams at the TREC-10 conference.

In the work of [AG00,LG98], answer formulations are produced specifically to improve web search engines. The formulations produced are precise, but they are used for query expansion to improve the retrieval of documents, not to retrieval of exact answers. While in [LG98] reformulation rules to transform a question like *What does NASDAQ stand for?* into `"NASDAQ stands for"` or `" NASDAQ means"` have been developed by hand, [AG00] uses machine learning to learn reformulation rules.

To our knowledge, however, answer formulation has not been investigated in the context of human-computer interaction (HCI) purposes to generate answer sentences rather than exact answers only. In the following, we will discuss our experiments in answer formulation for extraction and for HCI purposes, and argue that improving the linguistic quality of formulations is beneficial for both purposes.

3 Formulation Templates

Our first experiments with answer formulation were geared toward improving our results in answer extraction at the recent TREC-11 conference [VH02]. In this scenario, a high recall rate of the formulations was important in order to increase our chances of extracting the correct answer.

Because the TREC-11 questions are of general domain, we used the web as an additional source of information for answering questions and we used answer formulation to drive the search. That is, we searched the web for an exact phrase that could be the formulation of the answer to the question. For example, given the question *Who is the prime minister of Canada?*, our goal was to produce the formulation `"The prime minister of Canada is` <PERSON-NAME>`"`. Then, by searching the web for this exact phrase and extracting the noun phrase following it, our hope was to find the exact answer. Syntactic and semantic checks were then performed to ensure that the following noun phrase is indeed a PERSON-NAME. This prevented us from finding answers such as `"The prime minister of Canada is (a native of Shawinigan/very controversial/..."`).

To formulate an answer pattern from a question, we turn the latter into its declarative form using a set of hand-made patterns. We used the 200 questions of TREC-8 and the 693 questions of TREC-9 as training set to develop the formulation patterns and used the 500 questions of TREC-10 and the 500 questions of TREC-11 for testing.

Before the formulation is done, the question's grammatical form is normalized in order to restrict the number of cases to handle. For example, any question starting with *What's ...* is changed to *What is ...*, *What was the name of ...* is changed to *Name ...* In total, 17 grammatical rules are used for normalization. The formulation proper is then performed using a set of formulation templates

that test for the presence of specific keywords, grammatical tags and regular expressions. Figure 1 shows an example. The formulation template is composed of 2 sets of patterns: A question pattern that defines what the question must look like, and a set of answer patterns that defines a set of possible answer formulations. The patterns take into account specific keywords (eg. `When did`), strings of characters (`ANY-SEQUENCE-WORDS`) and part-of-speech tags (eg. `VERB-simple`). Answer patterns are specified using the same type of features plus a specification of the semantic class of the answer (eg. `TIME`). The semantic classes are used later, during answer extraction, to validate the nature of the candidate answers from the document. In the current implementation, about 10 semantic classes are used.

Formulation Template	**Example**
`When did ANY-SEQUENCE-WORDS-1 VERB-simple ?`	*(# 22) When did the Jurassic Period end?*
`ANY-SEQUENCE-WORDS-1 VERB-past TIME`	`the Jurassic Period ended TIME`
`TIME ANY-SEQUENCE-WORDS-1 VERB-past`	`TIME the Jurassic Period ended`
`TIME, ANY-SEQUENCE-WORDS-1 VERB-past`	`TIME, the Jurassic Period ended`

Figure 1. Example of a formulation template

A particular phenomenon that could not be dealt with using simple pattern-matching is the case of verb tenses. Many questions in the TREC collections are in the past tense; but the past tense is exhibited only in the auxiliary verb, while the main verb stays in its citation form. When formulating a declarative sentence, the tense information must be transferred to the main verb. In order to do this transformation, yet keep formulation rapid and straightforward, we extracted all the verbs from WordNet and built a hash table mapping their simple tense to their past tense.

To increase our chances of finding the exact answer, the formulation module can also generate conjunctions of formulations. For example, the question *(# 970) What type of currency is used in Australia?* is reformulated as <CLAUSE> `"is used in Australia"` AND <CLAUSE> `"is a type of currency"` where <CLAUSE> can be any string.

In total, 77 formulation templates are used. The templates are tried sequentially and all question patterns that are satisfied are activated. Table 1 shows the distribution of the templates by question type. For example, 6 templates can be used to transform *when*-type questions, and, on average, 1.7 answer formulations are produced for questions of that type. The 77 templates cover 93% of the 200 TREC-8 questions and 89.5% of the 693 TREC-9 questions. By coverage, we mean that at least one formulation template is applicable. The templates generate 412 formulations for the 186 processed TREC-8 questions and 1226

formulations for the 620 processed TREC-9 questions. So, on average, 2 answer formulations were produced per question.

Question Type	**Example**	**Nb of Templates**	**Average Nb of Answer Patterns**
when	*(# 398) When is Boxing Day?*	6	1.7
where	*(# 73) Where is the Taj Mahal?*	9	1.6
how many	*(# 214) How many hexagons are on a soccer ball?*	11	1.1
how much	*(# 203) How much folic acid should an expectant mother get daily?*	6	1.0
how (other)	*(# 177) How tall is Mt. Everest?*	11	1.4
what	*(# 257) What do penguins eat?*	21	1.0
which	*(# 108) Which company created the Internet browser Mosaic?*	2	1.0
who	*(# 55) Who started the Dominos Pizza chain?*	7	1.3
why	*(# 6) Why did David Koresh ask the FBI for a word processor?*	2	1.0
name	*(# 213) Name a flying mammal.*	2	1.0
Total		**77**	**1.2**

Table 1. Answer templates for each type of question

4 Evaluation

To evaluate the performance of the answer formulation module, we conducted three sets of experiments. The first was aimed at evaluating the answer formulation for answer extraction, the second experiment was meant to investigate HCI purposes, and the last was meant to evaluate how the linguistic quality of the formulation influences the score in answer extraction.

All experiments were performed on the TREC-10 and the TREC-11 question sets. The templates cover 448 of the 500 TREC-10 questions (89.6%) and 432 of the 500 TREC-11 questions (86.4%). This is shown in table 2. These numbers are consistent with the TREC-8 and TREC-9 questions used as training sets, and they are particularly high, especially considering that only 77 templates are used. In total, the templates generated 730 formulations for the 448 TREC-10 questions and 778 formulations for the TREC-11 questions. So, on average, 1.7 answer formulations were produced per question.

4.1 Evaluation for answer extraction

Evaluation for answer extraction was performed to specifically evaluate the improvement in answer extraction. To do so, we enhanced our QUANTUM QA sys-

Corpus	Nb questions	Coverage	Nb of formulations
TREC-8 (training)	200	93.0%	412
TREC-9 (training)	693	89.5%	1226
TREC-10 (testing)	500	89.6%	730
TREC-11 (testing)	500	86.9%	778

Table 2. Coverage of the formulation templates on different TREC question sets

tem [PK02] with the answer formulation module and used *Yahoo!* to search for web pages that contained the answer formulation. We then identified answer candidates by unification, and performed validity checks on candidates to ensure that the semantic class of the formulation was satisfied. Currently, semantic validation is rather simple and is based on the surface form of the candidates (eg. testing for length, capitalization, ...).

Only for 10% of the questions do we find one of the answer formulation in the TREC-10 document collection. However, when we search on the web, we find at least one occurrence of a formulation for 43% of the questions. Of these, the answer identified by unification is correct 51% of the time. In clear, 10 % of the TREC-9 and TREC-10 questions are correctly answered only by searching for answer formulations on the web and performing minimal semantic checking. For answer extraction, this simple technique of answer formulation seems interesting.

We further evaluated the answer formulation as part of the recent TREC-11 conference [VH02]. For 454 questions[1], without answer formulation, our system found 93 "good" answers[2] (20%). With answer formulation, our system found 110 "good" answers (24%). These results clearly show that using simple regular expressions based on keywords and part-of-speech tags can significantly improve answer extraction. Table 3 shows the percentage of good answers by question type on the TREC-11 corpus. *When*, *where*, *how-much* and *who*-type questions seem to benefit the most from answer formulation. We suspect that this is because declarative sentences introducing this type of information are more stereotypical; thus a small number of reformulation patterns are sufficient to cover a larger number of answers.

4.2 Evaluation for HCI purposes

To evaluate our answer formulation for HCI purposes, we generated answer formulations for the TREC-10 and TREC-11 questions. In total, 1510 answer formulations were generated for 1000 questions. We then asked 3 humans to judge

[1] 46 of the 500 TREC-11 questions were removed from the experiment because they had no answer in the TREC collection.

[2] By *good answer*, we mean an answer that is either correct, inexact with respect to its length or unsupported by its source document according to the NIST judgment. However, unlike for TREC, we consider in our evaluation all candidates tying for the best answer of a given question.

Question Type	% "good" answers without formulation	% "good" answers with formulation
when	14%	20%
where	29%	39%
how many	33%	33%
how much	10%	40%
how (other)	21%	18%
what	18%	22%
which	13%	13%
who	16%	31%
Total	**20%**	**24%**

Table 3. NIST-like judgment of answers produced by QUANTUM with and without answer formulations, for the TREC-11 questions (no-answer questions excluded)

these formulations on the basis of their grammaticality. The judgment could be one of the following:

type U (ungrammatical) The formulation is not grammatically correct. For example, `"it from Denver to Aspen is"` <DISTANCE> `"away"`.

type A (awkward) The formulation is grammatically correct for answer extraction but not a natural and responsive answer to the original question. For example, *(# 907) Who was the first man to fly across the Pacific Ocean?* ⇒ `"the first man to fly across the Pacific Ocean,"` <PERSON-NAME>.

type R (responsive) The formulation is grammatically correct and is natural and responsive to the original question. For example, *(# 914) Who was the first American to walk in space?* ⇒ `"the first American to walk in space was"` <PERSON-NAME>.

Inter-judge agreement was the following: 82% of the questions were judged similarly by all 3 judges; 18% of the questions were judged in two different categories by the 3 judges and 0% (1 out of 1508 answers) was judged differently by all judges.

Table 4 shows the results of the evaluation. On average, 56% of the formulations were considered correct for extraction as well as for HCI purposes; they were grammatical as well as natural and responsive to the question. 18% of the questions were considered awkward (appropriate for extraction, but not for HCI) and 19% were simply ungrammatical. Although the percentage of responsive formulations was higher than what we had originally expected, only one answer out of two being responsive is clearly unacceptable for human-computer interaction, where the end-users are humans, and more linguistically-motivated formulations are required.

Question Type	% without a formulation	Judgment %U	%A	%R
when	10	22	15	63
where	5	14	34	47
how many	29	43	0	29
how much	72	6	0	29
how (other)	20	14	6	58
what	6	27	9	58
which	10	61	29	0
who	0.5	4	37	59
why	100	0	0	0
name	0	0	0	100
Total	**7**	**19**	**18**	**56**

Table 4. Human judgment of answer formulations for HCI

4.3 Influence of the type of formulation for answer extraction

Finally, the last experiment that we performed was aimed at determining if the type of formulation (as determined for HCI purposes) has an influence on answer extraction. For example, do ungrammatical formulations really have no effect on answer extraction, or do they actually introduce noise and decrease the performance of extraction? If this is the case, then producing only good-quality formulations will be worth the effort not only for HCI, but also for answer extraction. To verify this, we evaluated answer extraction with 3 different sets of formulations:

type R: only formulations judged responsive by all 3 judges.
type R+A: formulations judged responsive or judged awkward by all 3 judges (same judgment by all judges).
type R+A+U: formulations judged responsive, awkward or ungrammatical by all 3 judges (same judgment by all judges).

Tables 5 and 6 show the result of this evaluation with the TREC-10 and TREC-11 corpora.

Corpus	Coverage		
	R	**R+A**	**R+A+U**
TREC-10	69.2%	73.8%	85.2%
TREC-11	58.4%	63.6%	79.2%

Table 5. Coverage of the formulation templates according to their formulation type

Table 5 shows that considering more formulation types covers more questions. However, table 6 shows that more formulation types does not result in

better quality of the extracted answers. As expected, considering responsive and awkward formulations (types R+A) yields the best score for answer extraction in both the TREC-10 and the TREC-11 question sets. Although the increase in score is slight when compared with taking only responsive formulations (28.3% versus 26.7%), it does correlate with our expectations and our definitions of a *responsive* answer and an *awkward* answer. Awkward formulations are therefore important for answer extraction. Considering ungrammatical answers (type U) has almost no effect on answer extraction and can actually introduce noise. In our experiment, ungrammatical formulations slightly decrease the score of answer extraction with both question sets (see table 6).

Corpus	**% good answers**		
	R	**R+A**	**R+A+U**
TREC-10	37.0%	38.3%	37.2%
TREC-11	16.5%	18.3%	17.4%
Total	**26.7%**	**28.3%**	**28.0%**

Table 6. Good answers found by QUANTUM, according to the responsiveness of the formulations

This last experiment shows that the linguistic quality of the formulations should be taken into account when used for QA. Although only responsive formulations should be generated for HCI purposes, responsive as well as awkward formulations should be used for answer extraction. For both purposes, ungrammatical formulations should not be used as they are not acceptable for HCI, and have no effect positive on answer extraction.

5 Conclusion and Future Work

In this paper, we have shown that simple hand-made patterns for answer formulation can greatly benefit answer extraction. In addition, the formulations that are generated in this manner are of better linguistic quality than brute force word permutations, and this allows us to add a human-computer interaction dimension to QA. We have also shown that generating only formulations of good linguistic quality not only is beneficial for HCI purposes, without decreasing the performance of answer extraction.

Our work has investigated only the case of individual questions. However, in a dialog-based human-computer interaction with a QA system, users will need to be able to ask a series of related and follow-up questions. Such contextual questions have already been taken into account in the context of QA [VH01], but we have not yet included them in our experiments. Dealing with them will lead to interesting issues such as question interpretation and dialog modeling.

As our work is based on the TREC question set, only fact-based questions were considered. These questions can be answered with noun-phrases, which

limits the scope of answer patterns. More complex types of questions such as *how?* that require a more complex answer have not been dealt with.

5.1 Acknowledgments

This project was financially supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) and the Bell University Laboratories (BUL).

References

[AG00] Eugene Agichtein and Luis Gravano. Snowball: Extracting relations from large plain-text collections. In *Proceedings of the 5th ACM International Conference on Digital Libraries*, 2000.

[BDB02] Eric Brill, Susan Dumais, and Michel Banko. An Analysis of the AskMSR Question-Answering System. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP-2002)*, Philadelphia, 2002.

[BLB+01] E. Brill, J. Lin, M. Banko, S. Dumais, and A. Ng. Data-Intensive Question Answering. In *Proceedings of The Tenth Text Retrieval Conference (TREC-X)*, pages 393–400, Gaithersburg, Maryland, 2001.

[CCL+01] C.L.A. Clarke, G.V. Cormack, T.R. Lynam, C.M. Li, and G.L. McLearn. Web Reinforced Question Answering (MultiText Experiments for TREC 2001). In *Proceedings of The Tenth Text Retrieval Conference (TREC-X)*, pages 673–679, Gaithersburg, Maryland, 2001.

[ea01] J. Burger et al. Issues, tasks and program structures to roadmap research in question & answering (q&a). Technical report, 2001. www-nlpir.nist.gov/projects/duc/roadmapping.html.

[HHL01] E. Hovy, U. Hermjakob, and C.-Y. Lin. The Use of External Knowledge in Factoid QA. In *Proceedings of The Tenth Text REtrieval Conference (TREC-X)*, pages 166–174, Gaithersburg, Maryland, 2001.

[HMP+01] S. Harabagiu, D. Moldovan, M. Pasca, R. Mihalcea, M. Surdeanu, R. Bunescu, R. Girju, V. Rus, and P. Morarescu. The role of lexico-semantic feedbacks in open-domain textual question answering. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics (ACL-2001)*, pages 274–281, Toulouse, France, July 2001.

[LCC01] T. Lynam, C. Clarke, and G. Cormack. Information extraction with term frequencies. In *Proceedings of HLT 2001 – First International Conference on Human Language Technology Research*, pages 169–172, San Diego, California, March 2001.

[LG98] Steve Lawrence and C. Lee Giles. Context and page analysis for improved web search. *IEEE Internet Computing*, 2(4):38–46, 1998.

[NIS00] NIST. *Proceedings of The Ninth Text REtrieval Conference (TREC-9)*, Gaithersburg, Maryland, 2000. available at http://trec.nist.gov/pubs/trec9/t9_proceedings.html.

[PK02] Luc Plamondon and Leila Koseim. Quantum: A function-based question answering system. In R. Cohen and B. Spencer, editors, *Proceedings of The Fifteenth Canadian Conference on Artificial Intelligence (AI'2002) - Lecture Notes in Artificial Intelligence no. 2338*, pages 281–292, Calgary, May 2002.

[VH99] E. M. Voorhees and D. K. Harman, editors. *Proceedings of The Eight Text REtrieval Conference (TREC-8)*, Gaithersburg, Maryland, November 1999. NIST. available at http://trec.nist.gov/pubs/trec8/t8_proceedings.html.

[VH01] E. M. Voorhees and D. K. Harman, editors. *Proceedings of The Tenth Text REtrieval Conference (TREC-X)*, Gaithersburg, Maryland, November 2001. NIST. available at http://trec.nist.gov/pubs/trec10/t10_proceedings.html.

[VH02] E. M. Voorhees and D. K. Harman, editors. *Proceedings of The Eleventh Text REtrieval Conference (TREC-11)*, Gaithersburg, Maryland, November 2002. NIST. to appear.

