

Downloaded from UvA-DARE, the institutional repository of the University of Amsterdam (UvA)  
<http://hdl.handle.net/11245/2.38045>

---

File ID	uvapub:38045
Filename	ecir2005-projection.pdf
Version	unknown

---

SOURCE (OR PART OF THE FOLLOWING SOURCE):

Type	book chapter
Title	Query Formulation for Answer Projection
Author(s)	G.A. Mishne, M. de Rijke
Faculty	FNWI: Informatics Institute (II)
Year	2005

FULL BIBLIOGRAPHIC DETAILS:

<http://hdl.handle.net/11245/1.241843>

---

*Copyright*

*It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content licence (like Creative Commons).*

---

# Query Formulation for Answer Projection

Gilad Mishne and Maarten de Rijke

Informatics Institute, University of Amsterdam  
Kruislaan 403, 1098 SJ Amsterdam, The Netherlands  
E-mail: `gilad,mdr@science.uva.nl`

**Abstract.** We examine the effects of various query modifications on the problem of answer projection — the task of retrieving documents that support a given answer to a question. We compare different techniques such as phrase searches and term weighting, and show that some models achieve significant improvements over unmodified queries.

## 1 Introduction

Largely spawned by the introduction of Question Answering (QA) tasks at evaluation forums such as TREC and CLEF, research on QA has intensified in recent years, with a strong focus on open-domain QA from a large collection of newspaper articles. As TREC and CLEF participants started moving to data-intensive (as opposed to knowledge-intensive) approaches, they discovered that for open-domain QA, consulting a much larger corpus — especially the web — often leads to improvements in performance. Efforts to move closer to shrink-wrapped trainable QA systems have brought with it a heavy usage of the web [1,4] and other external resources, including gazetteers, WordNet and others [3,5,8].

Overall, users prefer a paragraph-sized chunk of text over just an exact phrase as the answer to their questions, and they generally prefer answers embedded in context, regardless of the perceived reliability of the source documents [7]. For QA systems that locate answers and partial answers in external resources, this creates a new challenge of *answer projection*, i.e., of finding a *supporting document*, one from which a human can deduce that the answer is correct.<sup>1</sup>

This, then, is the answer projection task we address in the paper: given an answer to a question, find supporting documents in a given collection for it. Phrased this way, the task resembles a known-item search task. Accordingly, answer projection has been addressed using the kind of high precision retrieval models that have typically been employed for known item search, such as specific Okapi settings [2], passage retrieval, and combinations of heuristics [6]. Instead of varying the retrieval model, in this paper we adopt a basic vector space model, for which various query operators are well understood, and explore different query formulation techniques and their effect on the projection task.

---

<sup>1</sup> As an aside, by the guidelines for QA evaluations at TREC and CLEF, with each answer, a supporting document has to be identified in a given corpus. Failure to return a supporting document with an otherwise correct answer is a significant problem [2,5].

## 2 Query Formulation

In our experiments, the basic similarity between a document  $d$  and a query  $q$  is  $sim(q, d) = \sum_{t \in q} \frac{tf_{t,q} \cdot idf_t}{norm_q} \cdot \frac{tf_{t,d} \cdot idf_t}{norm_d} \cdot coord_{q,d} \cdot weight_t$ , where  $weight_t$  is a user-assigned term weight term,  $tf_{t,X} = \sqrt{(\text{freq}(t, X))}$ ,  $idf_t = 1 + \log(|D|/\text{freq}(t, D))$ , while  $norm_q = \sqrt{(\sum_{t \in q} tf_{t,q} \cdot idf_t^2)}$ ,  $norm_d = \sqrt{|d|}$  and  $coord_{q,d} = |q \cap d|/|q|$ . The terms in the query and the document may be phrases; in this case, the  $tf$  and  $idf$  scores for them are calculated accordingly. The following models were tested: **baseline** (the query is all words from the question and the answer), **boost-answer-N** (same as the baseline, but the answer words are weighted higher than other words, by a factor of N), **boolean-answer** (same as baseline, the answer words must be in the document), **phrases** (the query is all words from the question and answer; consecutive words that are phrases according to shallow analysis such as capitalization are phrased), and **phrase-answer** (all words from the question as single-word term, and the answer as a phrase). In addition, we consider combinations of the models, e.g., “**boolean-answer, boost-answer**” means that the answer words are required in the documents, and given higher term weights.

The models are based on simple “answer projection heuristics”: it is *likely* that if the answer contains more than one word, it is a phrase; it is *likely* that all words in the answer must be in the supporting document, while not necessarily all words from the question will be there; and so on.

## 3 Evaluation

To evaluate the effectiveness of the different query formulation methods for answer projection, we used a collection of 786 factoid questions taken from the QA tasks at TREC 11 and 12; these consist of all factoid questions having an answer in the AQUAINT collection, according to the judgment set released by NIST. A question may have more than a single answer, and an answer may have more than a single supporting document; for example, for question 2378. *How did Bob Marley die?*, both the answers “cancer” and “melanoma” are correct, and each is supported by more than one document. In total, there were 1814 correct answers to evaluate.

For each (question, answer)-pair we formulated queries according to the models presented in the previous section; we used standard stopping and Snowball stemming. To compare the models, we use both p@1 (“precision at rank 1”) and MRR (“Mean Reciprocal Rank”) measures; p@1 determines whether the top retrieved document is a supporting one, thus checking whether the method is useful in a real-life QA system, which looks only at the top retrieved document during the justification stage; the MRR score shows how good the model is in pushing supporting documents higher in the ranking.

The results of the comparison are listed in Figure 1. All results are strongly statistically significant (using the sign test), except those of the **phrases** method. Usage of phrase and boolean operators results in a clear, gradual increase in performance, and combinations of them improve results further. On the other hand, the simple term weighting used degrades performance; this can be attributed to

Model	MRR	p@1
baseline	0.477	0.346
boost-answer-2	0.464 (-3%)	0.340 (-1%)
boost-answer-5	0.408 (-14%)	0.287 (-17%)
boost-answer-20	0.329 (-31%)	0.225 (-35%)
phrases	0.471 (-1%)	0.347 (0%)
boolean-answer	0.502 (+5%)	0.374 (+8%)
phrase-answer	0.525 (+10%)	0.398 (+15%)
phrases,phrase-answer	0.517 (+8%)	0.397 (+15%)
phrases,phrase-answer,boolean-answer	<b>0.531 (+11%)</b>	<b>0.416 (+20%)</b>

Fig. 1. Comparison of the query formulation models

topic drift resulting from too much importance given to the answer, which is usually 1-2 terms and may have high *idf* values. We used shallow phrase recognition, and expect that deeper methods will improve the score further.

## 4 Conclusions

We address the answer projection task in Question Answering as a query formulation problem. Using a vector space model as a black box, we experiment with various methods of refining a query composed of the question, the answer, and various query operators as the building blocks. Our experiments show a consistent and significant improvement for some models and their combinations.

**Acknowledgments.** Both authors were supported by the Netherlands Organization for Scientific Research (NWO) under project number 220-80-001. De Rijke was also supported by NWO under project numbers 365-20-005, 612.069.-006, 612.000.106, 612.000.207, 612.066.302, 264-70-050, and 017.001.190.

## References

1. E. Brill, S. Dumais, and M. Banko. An analysis of the AskMSR question-answering system. In *Proc. 39th Annual ACL*, 2002.
2. E. Brill, J. Lin, M. Banko, S.T. Dumais, and A.Y. Ng. Data-intensive question answering. In *Proc. 10th Text REtrieval Conference*, 2001.
3. C. Clarke, G. Cormack, G. Kemkes, M. Laszlo, T. Lynam, E. Terra, and P. Tilker. Statistical selection of exact answers. In *Proc. 11th Text REtrieval Conference*, 2002.
4. A. Ittycheriah, M. Franz, and S. Roukos. IBM's statistical question answering system. In *Proc. 10th Text REtrieval Conference*, 2001.
5. V. Jijkoun, G. Mishne, and M. de Rijke. How frogs built the Berlin Wall. In *Proc. CLEF 2003*. Springer, 2004.
6. J. Lin, A. Fernandes, B. Katz, G. Marton, and S. Tellex. Extracting answers from the web using knowledge annotation and knowledge mining techniques. In *Proc. 11th Text REtrieval Conference*, 2002.
7. J. Lin, D. Quan, V. Sinha, K. Bakshi, D. Huynh, B. Katz, and D.R. Karger. The role of context in question answering systems. In *CHI '03 Extended Abstracts on Human Factors in Computing Systems*. ACM Press, 2003.
8. L.V. Lita, W. Hunt, and E. Nyberg. Resource analysis for question answering. In *Proc. 42nd Annual ACL*, 2004.