What Is Not in the Bag of Words for *Why*-QA?

Suzan Verberne*
Radboud University Nijmegen

Lou Boves**
Radboud University Nijmegen

Nelleke Oostdijk†
Radboud University Nijmegen

Peter-Arno Coppen‡
Radboud University Nijmegen

While developing an approach to why*-QA, we extended a passage retrieval system that uses off-the-shelf retrieval technology with a re-ranking step incorporating structural information. We get significantly higher scores in terms of MRR@150 (from 0.25 to 0.34) and success@10. The 23% improvement that we reach in terms of MRR is comparable to the improvement reached on different QA tasks by other researchers in the field, although our re-ranking approach is based on relatively lightweight overlap measures incorporating syntactic constituents, cue words, and document structure.*

1. Introduction

About 5% of all questions asked to QA systems are *why*-questions (Hovy, Hermjakob, and Ravichandran 2002). *Why*-questions need a different approach than factoid questions, because their answers are explanations that usually cannot be stated in a single phrase. Recently, research (Verberne 2006; Higashinaka and Isozaki 2008) has been directed at QA for *why*-questions (*why*-QA). In earlier work on answering *why*-questions on the basis of Wikipedia, we found that the answers to most *why*-questions are passages of text that are at least one sentence and at most one paragraph in length (Verberne et al. 2007b). Therefore, we aim at developing a system that takes as input a *why*-question and gives as output a ranked list of candidate answer passages.

In the current article, we propose a three-step setup for a *why*-QA system: (1) a question-processing module that transforms the input question to a query; (2) an off-the-shelf retrieval module that retrieves and ranks passages of text that share content

* Department of Linguistics, PO Box 9103, 6500 HD Nijmegen, the Netherlands. E-mail: s.verberne@let.ru.nl.
** Department of Linguistics, PO Box 9103, 6500 HD Nijmegen, the Netherlands. E-mail: l.boves@let.ru.nl.
† Department of Linguistics, PO Box 9103, 6500 HD Nijmegen, the Netherlands. E-mail: n.oostdijk@let.ru.nl.
‡ Department of Linguistics, PO Box 9103, 6500 HD Nijmegen, the Netherlands. E-mail: p.a.coppen@let.ru.nl.

Submission received: 30 July 2008; revised submission received: 18 February 2009; accepted for publication: 4 September 2009.

with the input query; and (3) a re-ranking module that adapts the scores of the retrieved passages using structural information from the input question and the retrieved passages.

In the first part of this article, we focus on step 2, namely, passage retrieval. The classic approach to finding passages in a text collection that share content with an input query is retrieval using a bag-of-words (BOW) model (Salton and Buckley 1988). BOW models are based on the assumption that text can be represented as an unordered collection of words, disregarding grammatical structure. Most BOW-based models use statistical weights based on term frequency, document frequency, passage length, and term density (Tellex et al. 2003).

Because BOW approaches disregard grammatical structure, systems that rely on a BOW model have their limitations in solving problems where the syntactic relation between words or word groups is crucial. The importance of syntax for QA is sometimes illustrated by the sentence *Ruby killed Oswald*, which is not an answer to the question *Who did Oswald kill?* (Bilotti et al. 2007). Therefore, a number of researchers in the field investigated the use of structural information on top of a BOW approach for answer retrieval and ranking (Tiedemann 2005; Quarteroni et al. 2007; Surdeanu, Ciaramita, and Zaragoza 2008). These studies show that although the BOW model makes the largest contribution to the QA system results, adding structural (syntactic information) can give a significant improvement.

In the current article, we hypothesize that for the relatively complex problem of *why*-QA, a significant improvement—at least comparable to the improvement gained for factoid QA—can be gained from the addition of structural information to the ranking component of the QA system. We first evaluate a passage retrieval system for *why*-QA based on standard BOW ranking (step 1 and 2 in our set-up). Then we perform an analysis of the strengths and weaknesses of the BOW model for retrieving and ranking candidate answers. In view of the observed weaknesses of the BOW model, we choose our feature set to be applied to the set of candidate answer passages in the re-ranking module (step 3 in our set-up).

The structural features that we propose are based on the idea that some parts of the question and the answer passage are more important for relevance ranking than other parts. Therefore, our re-ranking features are overlap-based: They tell us which parts of a *why*-question and its candidate answers are the most salient for ranking the answers. We evaluate our initial and adapted ranking strategies using a set of *why*-questions and a corpus of Wikipedia documents, and we analyze the contribution of both the BOW model and the structural features.

The main contributions of this article are: (1) we address the relatively new problem of *why*-QA and (2) we analyze the contribution of overlap-based structural information to the problem of answer ranking.

The paper is organized as follows. In Section 2, related work is discussed. Section 3 presents the BOW-based passage retrieval method for *why*-QA, followed by a discussion of the strengths and weaknesses of the approach in Section 4. In Section 5, we extend our system with a re-ranking component based on structural overlap features. A discussion of the results and our conclusions are presented in Sections 6 and 7, respectively.

2. Related Work

We distinguish related work in two directions: research into the development of systems for *why*-QA (Section 2.1), and research into combining structural and BOW features for QA (Section 2.2).

2.1 Research into *Why*-QA

In related work (Verberne et al. 2007a), we focused on selecting and ranking explanatory passages for *why*-QA with the use of rhetorical structures. We developed a system that employs the discourse relations in a manually annotated document collection: the RST Treebank (Carlson, Marcu, and Okurowski 2003). This system matches the input question to a text span in the discourse tree of the document and it retrieves as answer the text span that has a specific discourse relation to this question span. We evaluated our method on a set of 336 *why*-questions formulated to seven texts from the WSJ corpus. We concluded that discourse structure can play an important role in *why*-QA, but that systems relying on these structures can only work if candidate answer passages have been annotated with discourse structure. Automatic parsers for creating full rhetorical structures are currently unavailable. Therefore, a more practical approach appears to be necessary for work in *why*-QA, namely, one which is based on automatically created annotations.

Higashinaka and Isozaki (2008) focus on the problem of ranking candidate answer paragraphs for Japanese *why*-questions. They assume that a document retrieval module has returned the top 20 documents for a given question. They extract features for content similarity, causal expressions, and causal relations from two annotated corpora and a dictionary. Higashinaka and Isozaki evaluate their ranking method using a set of 1,000 *why*-questions that were formulated to a newspaper corpus by a text analysis expert. 70.3% of the reference answers for these questions are ranked in the top 10 by their system, and MRR[1] was 0.328.

Although the approach of Higashinaka and Isozaki is very interesting, their evaluation collection has the same flaw as the one used by Verberne et al. (2007a): Both collections consist of questions formulated to a pre-selected answer text. Questions elicited in response to newspaper texts tend to be unrepresentative of questions asked in a real QA setting. In the current work, therefore, we work with a set of questions formulated by users of an online QA system (see Section 3.1).

2.2 Combining Structural and Bag-of-Words Features for QA

Tiedemann (2005) investigates syntactic information from dependency structures in passage retrieval for Dutch factoid QA. He indexes his corpus at different text layers (BOW, part-of-speech, dependency relations) and uses the same layers for question analysis and query creation. He optimizes the query parameters for the passage retrieval task by having a genetic algorithm apply the weights to the query terms. Tiedemann finds that the largest weights are assigned to the keywords from the BOW layer and to the keywords related to the predicted answer type (such as 'person'). The baseline approach, using only the BOW layer, gives an MRR of 0.342. Using the optimized IR settings with additional layers, MRR improves to 0.406.

Quarteroni et al. (2007) consider the problem of answering definition questions. They use predicate–argument structures (PAS) for improved answer ranking. They find that PAS as a stand-alone representation is inferior to parse tree representations, but that together with the BOW it yields higher accuracy. Their results show a significant

1 The reciprocal rank (RR) for a question is 1 divided by the rank ordinal of the highest ranked relevant answer. The Mean RR is obtained by averaging RR over all questions.

improvement of PAS–BOW compared to parse trees (F-scores 70.7% vs. 59.6%) but PAS makes only a very small contribution compared to BOW only (which gives an F-score of 69.3%).

Recent work by Surdeanu, Ciaramita, and Zaragoza (2008) addresses the problem of answer ranking for *how-to*-questions. From Yahoo! Answers,[2] they extract a corpus of 140,000 answers with 40,000 questions. They investigate the usefulness of a large set of question and answer features in the ranking task. They conclude that the linguistic features "yield a small, yet statistically significant performance increase on top of the traditional BOW and *n*-gram representation (page 726)."

All these authors conclude that the addition of structural information in QA gives a small but significant improvement compared to using a BOW-model only. For *why*-questions, we also expect to gain improvement from the addition of structural information.

3. Passage Retrieval for *Why*-QA Using a BOW Model

As explained in Section 1, our system comprises three modules: *question2query*, passage retrieval, and re-ranking. In the current section, we present the first two system modules, and the re-ranking module, including a description of the structural features that we consider, is presented in Section 5. First, however, we describe our data collection and evaluation method.

3.1 Data and Evaluation Set-up

For our experiments, we use the Wikipedia INEX corpus (Denoyer and Gallinari 2006). This corpus consists of all 659,388 articles from the online Wikipedia in the summer of 2006 in XML format.

For development and testing purposes, we exploit the Webclopedia question set (Hovy, Hermjakob, and Ravichandran 2002), which contains questions asked to the online QA system *answers.com*. Of these questions, 805 (5% of the total set) are *why*-questions. For 700 randomly selected *why*-questions, we manually searched for an answer in the Wikipedia XML corpus, saving the remaining 105 questions for future testing purposes. 186 of these 700 questions have an answer in the corpus.[3] Extraction of one relevant answer for each of these questions resulted in a set of 186 *why*-questions and their reference answers.[4] Two examples illustrate the type of data we are working with:

1. "Why didn't Socrates leave Athens after he was convicted?" — "Socrates considered it hypocrisy to escape the prison: he had knowingly agreed to live under the city's laws, and this meant the possibility of being judged guilty of crimes by a large jury."

2 See `http://answers.yahoo.com/`.

3 Thus, about 25% of our questions have an answer in the Wikipedia corpus. The other questions are either too specific (*Why do ceiling fans turn counter-clockwise but table fans turn clockwise?*) or too trivial (*Why do hotdogs come in packages of 10 and hotdog buns in packages of 8?*) for the coverage of Wikipedia in 2006.

4 Just like factoid questions, most *why*-questions generally have one correct answer that can be formulated in different ways.

2. "Why do most cereals crackle when you add milk?" — "They are made of a sugary rice mixture which is shaped into the form of rice kernels and toasted. These kernels bubble and rise in a manner which forms very thin walls. When the cereal is exposed to milk or juices, these walls tend to collapse suddenly, creating the famous 'Snap, crackle and pop' sounds."

To be able to do fast evaluation without elaborate manual assessments, we manually created one answer pattern for each of the questions in our set. The answer pattern is a regular expression that defines which of the retrieved passages are considered a relevant answer to the input question. The first version of the answer patterns was directly based on the corresponding reference answer, but in the course of the development and evaluation process, we extended the patterns in order to cover as many as possible of the Wikipedia passages that contain an answer. For example, for question 1, we developed the following answer pattern based on two variants of the correct answer that occur in the corpus: `/(Socrates.* opportunity.* escape.* Athens.* considered.* hypocrisy | leave.* run.* away.* community.* reputation)/`.[5]

In fact, answer judgment is a complex task due to the presence of multiple answer variants in the corpus. It is a time-consuming process because of the large number of candidate answers that need to be judged when long lists of answers are retrieved per question. In future work, we will come back to the assessment of relevant and irrelevant answers.

After applying our answer patterns to the passages retrieved, we count the questions that have at least one relevant answer in the top n results. This number divided by the total number of questions in a test set gives the measure *success@n*. In Section 3.2, we explain the levels for n that we use for evaluation. For the highest ranked relevant answer per question, we determine the RR. Questions for which the system did not retrieve an answer in the list of 150 results get an RR of 0. Over all questions, we calculate the mean reciprocal rank MRR.

3.2 Method and Results

In the *question2query* module of our system we convert the input question to a query by removing stop words[6] and punctuation, and simply list the remaining content words as query terms.

The second module of our system performs passage retrieval using off-the-shelf retrieval technology. In Khalid and Verberne (2008), we compared a number of settings for our passage retrieval task. We considered two different retrieval engines (Lemur[7] and Wumpus[8]), four different ranking models, and two types of passage segmentation: disjoint and sliding passages. In each setting, 150 results were obtained by the retrieval engine and ranked by the retrieval model. We evaluated all retrieval settings in terms of

5 Note that the vertical bar separates the two alternatives.

6 To this end we use the stop word list that can be found at `http://marlodge.supanet.com/museum/funcword.html`. We use all items except the numbers and the word *why*.

7 Lemur is an open source toolkit for information retrieval that provides flexible support for different types of retrieval models. See `http://www.lemurproject.org`.

8 Wumpus is an information retrieval system mainly geared at XML retrieval. See `http://www.wumpus-search.org/`.

MRR@n[9] and success@n for levels $n = 10$ and $n = 150$. For the evaluation of the retrieval module, we were mainly interested in the scores for success@150 because re-ranking can only be successful if at least one relevant answer was returned by the retrieval module.

We found that the best-scoring passage retrieval setting in terms of success@150 is Lemur on an index of sliding passages with TF-IDF (Zhai 2001) as ranking model. We obtained the following results with this passage retrieval setting: success@150 is 78.5%, success@10 is 45.2%, and MRR@150 is 0.25. We do not include the results obtained with the other retrieval settings here because the differences were small.

The results show that for 21.5% of the questions in our set, no answer was retrieved in the top-150 results. We attempted to increase this coverage by retrieving 250 or 500 answers per question but this barely increased the success score at maximum n. The main problems for the questions that we miss are infamous retrieval problems such as the vocabulary gap between a question and its answer. For example, the answer to *Why do chefs wear funny hats?* contains none of the words from the question.

4. The Strengths and Weaknesses of the BOW Model

In order to understand how answer ranking is executed by the passage retrieval module, we first take a closer look at the TF-IDF algorithm as it has been implemented in Lemur. TF-IDF is a pure BOW model: Both the query and the passages in the corpus are represented by the term frequencies (numbers of occurrences) for each of the words they contain. The terms are weighted using their inverse document frequency (IDF), which puts a higher weight on terms that occur in few passages than on terms that occur in many passages. The term frequency (TF) functions for the query and the document, and the parameter values chosen for these functions in Lemur can be found in Zhai (2001).

As explained in the previous section, we consider success@150 to be the most important measure for the retrieval module of our system. However, for the system as a whole, success@10 is a more important evaluation measure. This is because users tend to pay much more attention to the top 10 results of a retrieval system than to results that are ranked lower (Joachims et al. 2005). Therefore, it is interesting to investigate which questions are answered in the top 150 and not in the top 10 by our passage retrieval module. This is the set of questions for which the BOW model is not effective enough and additional (more specific) overlap information is needed for ranking a relevant answer in the top 10.

We analyzed the set of questions that get a relevant answer at a rank between 10 and 150 (62 questions), which below we will refer to as our **focus set**. We compared our focus set to the questions for which a relevant answer is in the top 10 (84 questions). Although these numbers are too small to do a quantitative error analysis, a qualitative analysis provides valuable insights into the strengths and weaknesses of a BOW representation such as TF-IDF. In Sections 4.1 to 4.4 we discuss four different aspects of *why*-questions that present problems for the BOW model.

9 Note that MRR is often used without the explicit cut-off point (n). We add it to clarify that RR is 0 for the questions without a correct answer in the top-n.

4.1 Short Questions

Ten questions in our focus set contain only one or two content words. We can see the effect of short queries if we compare three questions that contain only one semantically rich content word.[10] The rank of the highest ranked relevant answer is given between parentheses; the last of these three questions is in our focus set.

1. Why do people hiccup? (2)
2. Why do people sneeze? (4)
3. Why do we dream? (76)

We found that the rank of the relevant answer is related to the corpus frequency of the single semantically rich word, which is 64 for *hiccup*, 220 for *sneeze*, and 13,458 for *dream*. This means that many passages are retrieved for question 3, making the chances for the relevant answer to be ranked in the top 10 smaller. One way to overcome the problem of long result lists for short queries is by adding words to the query that make it more specific. In the case of *why*-QA, we know that we are not simply searching for information on *dreaming* but for an *explanation* for *dreaming*. Thus, in the ranking process, we can extend the query with explanatory cue words such as *because*.[11] We expect that the addition of explanatory cue phrases will give an improvement in ranking performance.

4.2 The Document Context of the Answer

There are many cases where the context of the candidate answer gives useful information. Consider, for example, the question *Why does a snake flick out its tongue?*, the correct answer to which was ranked 29. A human searcher expects to find the answer in a Wikipedia article about *snakes*. Within the *Snake* article he or she may search for the words *flick* and/or *tongue* in order to find the answer. This suggests that in some cases there is a direct relation between a specific part of the question and the context (document and/or section) of the candidate answer. In cases like this, the answer document and the question apparently share the same topic (*snake*). By analogy with linguistically motivated approaches to factoid QA (Ferret et al. 2002), we introduce the term **question focus** for this topic.

In the example question *flick* is the word with the lowest corpus frequency (556), followed by *tongue* (4,925) and *snake* (6,809). Using a BOW approach to document title matching, candidate answers from documents with *flick* or *tongue* in their title would be ranked higher than answers from documents with *snake* in their title. Thus, for questions for which there is overlap between the question focus and the title of the answer documents (two thirds of the questions in our set), we can improve the ranking of candidate answers by correctly predicting the question focus. In Section 5.1.2, we make concrete suggestions for achieving this.

10 The word *people* in subject position is a semantically poor content word.

11 The addition of cue words can also be considered to be applied in the retrieval step. We come back to this in Section 6.3.

4.3 Multi-Word Terms

A very important characteristic of the BOW model is that words are considered separate terms. One of the consequences is that multi-word terms such as multi-word noun phrases (mwNPs) are not treated as a single term. Here, three examples of questions are shown in which the subject is realized by a mwNP (underlined in the examples; the rank of the relevant answer is shown between brackets):

1. Why are hush puppies called hush puppies? (1)
2. Why is the coral reef disappearing? (29)
3. Why is a black hole black? (31)

We investigated the corpus frequencies for the separate parts of each mwNP. We found that these are quite high for *coral* (3,316) and *reef* (2,597) compared to the corpus frequency of the phrase *coral reef* (365). The numbers are even more extreme for *black* (103,550) and *hole* (9,734) versus *black hole* (1,913). On the other hand, the answer to the *hush puppies* question can more easily be ranked because the corpus frequencies for the separate terms *hush* (594) and *puppies* (361) are relatively low. This shows that multi-word terms do not necessarily give problems for the BOW model as long as the document frequencies for the constituent words are relatively low. If (one of) the words in the phrase is/are frequent, it is very difficult to rank the relevant answer high in the result list with use of word overlap only.

In our focus set, 36 of the 62 questions contain a mwNP. For these questions, we can expect improved ranking from the addition of NPs to our feature set.

4.4 Syntactic Structure

The BOW model does not take into account sentence structure. The potential importance of sentence structure for improved ranking can be exemplified by the following two questions from our set. Note that both examples contain a subordinate clause (finite or non-finite):

1. Why do baking soda and vinegar explode when you mix them together? (4)
2. Why are there 72 points to the inch when discussing fonts and printing? (36)

In both cases, the contents of the subordinate clause are less important to the goal of the question than the contents of the main clause. In the first example, this is (coincidentally) reflected by the corpus frequencies of the words in both clauses: *mix* (12,724) and *together* (83,677) have high corpus frequencies compared to *baking* (832), *soda* (1,620), *vinegar* (871), and *explode* (1,285). As a result, the reference answer containing these terms is ranked in the top-10 by TF-IDF. In the second example, however, the corpus frequencies do not reflect the importance of the terms. *Fonts* and *printing* have lower corpus frequencies (1,243 and 6,978, respectively) than *points* (43,280) and *inch* (10,046). Thus, *fonts* and *printing* are weighted heavier by TF-IDF although these terms are only peripheral to the goal of the query, the core of which is *Why are there 72 points to the inch?* This cannot be derived from the corpus frequencies, but can only be inferred from the syntactic function (adverbial) of *when discussing fonts and printing* in the question.

Thus, the lack of information about sentence structure in the BOW model does not necessarily give rise to problems as long as the importance of the question terms is reflected by their frequency counts. If term importance does not align with corpus

frequency, grammatical structure becomes potentially useful. Therefore, we expect that syntactic structure can make a contribution to cases where the importance of the terms is not reflected by their corpus frequencies but can be derived from their syntactic function.

4.5 What Can We Expect from Structural Information?

In Sections 4.1 to 4.4 we discussed four aspects of *why*-questions that are problematic for the BOW model. We expect contributions from the inclusion of information on cue phrases, question focus and the document context of the answer, noun phrases, and the syntactic structure of the question. We think that it is possible to achieve improved ranking performance if features based on structural overlap are taken into account instead of global overlap information.

5. Adding Overlap-Based Structural Information

From our analyses in Section 4, we found a number of question and answer aspects that are potentially useful for improving the ranking performance of our system. In this section, we present the re-ranking module of our system. We define a feature set that is inspired by the findings from Section 4 and aims to find out which structural features of a question–answer pair contribute the most to better answer ranking. We aim to weigh these features in such a way that we can optimize ranking performance. The input data for our re-ranking experiments is the output of the passage retrieval module. A success@150 score of 78.5% for passage retrieval (see Section 3.2) means that the maximum success@10 score that we can achieve by re-ranking is 78.5%.

5.1 Features for Re-ranking

The first feature in our re-ranking method is the score that was assigned to a candidate answer by Lemur/TF-IDF in the retrieval module (f0). In the following sections we introduce the other features that we implemented. Each feature represents the overlap between two item bags:[12] a bag of question items (for example: all the question's noun phrases, or the question's main verb) and a bag of answer items (for example: all answer words, or all verbs in the answer). The value that is assigned to a feature is a function of the overlap between these two bags. We used the following overlap function:

$$S(Q,A) = \frac{Q_A + A_Q}{Q + A} \tag{1}$$

in which Q_A is the number of question items that occur at least once in the bag of answer items, A_Q is the number of answer items that occur at least once in the bag of question items, and $Q + A$ is the number of items in both bags of items joined together.

5.1.1 The Syntactic Structure of the Question. In Section 4.4, we argued that some syntactic parts of the question may be more important for answer ranking than others. Because we have no quantitative evidence yet which syntactic parts of the question are the most important, we created overlap features for each of the following question parts: phrase

12 Note that a "bag" is a set in which duplicates are counted as distinct items.

heads (f1), phrase modifiers (f2); the subject (f3), main verb (f4), nominal predicate (f5), and direct object (f6) of the main clause; and all noun phrases (f11). For each of these question parts, we calculated its word overlap with the bag of all answer words. For the features f3–f6, we added a variant where as answer items only words/phrases with the same syntactic function as the question token were included (f7, f8, f9, f10).

Consider for example question 1 from Section 3.1: *Why didn't Socrates leave Athens after he was convicted?*, and the reference answer as the candidate answer for which we are determining the feature values: *Socrates considered it hypocrisy to escape the prison: he had knowingly agreed to live under the city's laws, and this meant the possibility of being judged guilty of crimes by a large jury.*

From the parser output, our feature extraction script extracts *Socrates* as subject, *leave* as main verb, and *Athens* as direct object. Neither *leave* nor *Athens* occur in the answer passage, thus f4, f6, f8, and f10 are all given a value of 0. So are f5 and f9, because the question has no nominal predicate. For the subject *Socrates*, our script finds that it occurs once in the bag of answer words. The overlap count for the feature f3 is thus calculated as $\frac{1+1}{1+18} = 0.105$.[13] For the feature f7, our script extracts the grammatical subjects *Socrates*, *he*, and *this* from the parser's representation of the answer passage. Because the bag of answer subjects for f7 contains three items, the overlap is calculated as $\frac{1+1}{1+3} = 0.5$.

5.1.2 The Semantic Structure of the Question. In Section 4.2, we saw that often there is a link between the question focus and the title of the document in which the reference answer is found. In those cases, the answer document and the question share the same topic. For most questions, the focus is the syntactic subject: *Why do* cats *sleep so much?* Judging from our data, there are two exceptions to this general rule: (1) If the subject is semantically poor, the question focus is the (verbal or nominal) predicate: *Why do people* sneeze*?*, and (2) in case of etymology questions (which cover about 10% of *why*-questions), the focus is the subject complement of the passive sentence: *Why are chicken wings called* Buffalo Wings*?*

We included a feature (f12) for matching words from the question focus to words from the document title and a feature (f13) for the relation between question focus words and all answer words. We also include a feature (f14) for the other, non-focus question words.

5.1.3 The Document Context of the Answer. Not only is the document title in relation to the question focus potentially useful for answer ranking, but also other aspects of the answer context. We include four answer context features in our feature set: overlap between the question words and the title of the Wikipedia document (f15), overlap between question words and the heading of the answer section (f16), the relative position of the answer passage in the document (f17), and overlap between a fixed set of words that we selected as explanatory cues when they occur in a section heading and the set of words that occur in the section heading of the passage (f18).[14]

13 The bag of question subjects contains one item (*Socrates*, the 1 in the denominator) and one item from this bag occurs in the bag of answer words (the left 1 in the numerator). Without stopwords, the bag of all answer words contains 18 items, one of which occurs in the bag of question subjects (the right 1 in the numerator).

14 We found these section heading cues by extracting all section headings from the Wikipedia corpus, sorting them by frequency, and then manually marking those section heading words that we expect to occur with explanatory sections. The result is a small set of heading cues (*history*, *origin*, *origins*, *background*, *etymology*, *name*, *source*, *sources*) that is independent of the test set we work with.

5.1.4 Synonyms. For each of the features f1 to f10 and f12 to f16 we add an alternative feature (f19 to f34) covering the set of all WordNet synonyms for all question terms in the original feature. For synonyms, we apply a variant of Equation (1) in which Q_A is interpreted as the number of question items that have at least one synonym in the bag of answer items and A_Q as the number of answer items that occur in at least one of the synonym sets of the question items.

5.1.5 WordNet Relatedness. Additionally, we included a feature representing the relatedness between the question and the candidate answer using the WordNet Relatedness tool (Pedersen, Patwardhan, and Michelizzi 2004) (f35). As a measure of relatedness, we choose the Lesk measure, which incorporates information from WordNet glosses.

5.1.6 Cue Phrases. Finally, as proposed in Section 4.1, we added a closed set of cue phrases that are used to introduce an explanation (f36). We found these explanatory phrases in a way that is commonly used for finding answer cues and that is independent of our own set of question–answer pairs. We queried the key answer words to the most frequent *why*-question on the Web *Why is the sky blue?* (*blue sky rayleigh scattering*) to the MSN Search engine[15] and crawled the first 250 answer fragments that are retrieved by the engine. From these, we manually extracted all phrases that introduce the explanation. This led to a set of 47 cue phrases such as *because*, *as a result of*, *which explains why*, and so on.

5.2 Extracting Feature Values from the Data

For the majority of features we needed the syntactic structure of the input question, and for some of the features also of the answer. We experimented with two different syntactic parsers for these tasks: the Charniak parser (Charniak 2000) and a development version of the Pelican parser.[16] Of these, Pelican has a more detailed descriptive model and gives better accuracy but Charniak is at present more robust for parsing long sentences and large amounts of text. We parsed the questions with Pelican because we need accurate parsings in order to correctly extract all constituents. We parsed all answers (186 $\times$ 150 passages) with Charniak because of its speed and robustness.

For feature extraction, we used the following external components: A stop word list,[17] the sets of cue phrases as described in Sections 5.1.3 and 5.1.6, the CELEX Lemma lexicon (Burnage et al. 1990), the WordNet synonym sets, the WordNet Similarity tool (Pedersen, Patwardhan, and Michelizzi 2004), and a list of pronouns and semantically poor nouns.[18] We used a Perl script for extracting feature values for each question–answer pair. For each feature, the script composes the required bags of question items and answer items. All words are lowercased and punctuation is removed. For terms in the question set that consist of multiple words (for example, a multi-word subject), spaces are replaced by underscores before stop words are removed from the question and the answer. Then the script calculates the similarity between the two sets for each feature following Equation (1).[19]

15 `http://www.live.com`.
16 See `http://lands.let.ru.nl/projects/pelican/`.
17 See Section 3.1.
18 Semantically poor nouns that we came across in our data set are the nouns *humans* and *people*.
19 A multi-word term from the question is counted as one item.

Table 1
Results for the *why*-QA system: the complete system including re-ranking compared against plain Lemur/TF-IDF for 187 *why*-questions.

	Success@10	Success@150	MRR@150
Lemur/TF-IDF–sliding	45.2%	78.5%	0.25
TF-IDF + Re-ranking using 37 structural features	57.0%	78.5%	0.34

Whether or not to lemmatize the terms before matching them is open to debate. In the literature, there is some discussion on the benefit of lemmatization for question answering (Bilotti, Katz, and Lin 2004). Lemmatization can be especially problematic in the case of proper names (which are not always recognizable by capitalization). Therefore, we decided only to lemmatize verbs (for features f4 and f8) in the current version of our system.

5.3 Re-ranking Method

Feature extraction led to a vector consisting of 37 feature values for each of the 27,900 items in the data set. We normalized the feature values over all 150 answer candidates for the same question to a number between 0 and 1 using the L1 vector norm. Each instance (representing one question–answer pair) was automatically labeled 1 if the candidate answer matched the answer pattern for the question and 0 if it did not. On average, a *why*-question had 1.6 correct answers among the set of 150 candidate answers.

In the process of training our re-ranking module, we aim at combining the 37 features in a ranking function that is used for re-ordering the set of candidate answers. The task of finding the optimal ranking function for ranking a set of items is referred to as "learning to rank" in the information retrieval literature (Liu et al. 2007). In Verberne et al. (2009), we compared several machine learning techniques[20] for our learning-to-rank problem. We evaluated the results using 5-fold cross validation on the question set.

5.4 Results from Re-ranking

The results for the complete system compared with passage retrieval with Lemur/TF-IDF only are in Table 1. We show the results in terms of success@10, success@150, and MRR@150. We only present the results obtained using the best-performing learning-to-rank technique: logistic regression.[21] A more detailed description of our machine learning method and a discussion of the results obtained with other learning techniques can be found in Verberne et al. (2009).

20 Naive Bayes, Support Vector Classification, Support Vector Regression, Logistic regression, Ranking SVM, and a genetic algorithm, all with several optimization functions.

21 We used the *lrm* function from the Design package in *R* (`http://cran.r-project.org/web/packages/Design`) for training and evaluating models based on logistic regression.

After applying our re-ranking module, we found a significant improvement over bare TF-IDF in terms of success@10 and MRR@150 ($z = -4.29, p < 0.0001$ using the Wilcoxon Signed-Rank test for paired reciprocal ranks).

5.5 Which Features Made the Improvement?

In order to evaluate the importance of our features, we rank them according to the coefficient that was assigned to them in the logistic regression model (See Table 2). We only consider features that are significant at the $p = 0.05$ level. We find that all eight significant features are among the top nine features with the highest coefficient.

The feature ranking is discussed in Section 6.1.

6. Discussion

In the following sections, we discuss the feature ranking (Section 6.1), make a comparison to other re-ranking approaches (Section 6.2), and explain the attempts that we made at solving the remaining problems (Section 6.3).

6.1 Discussion of the Feature Ranking

Table 2 shows that only a small subset (8) of our 37 features significantly contribute to the re-ranking score. The highest ranked feature is TF-IDF (the bag of words), which is not surprising since TF-IDF alone already reaches an MRR@150 of 0.25 (see Section 3.2). In Section 4.5, we predicted a valuable contribution from the addition of cue phrases, question focus, noun phrases, and the document context of the answer. This is partly confirmed by Table 2, which shows that among the significant features are the feature that links question focus to document title and the cue phrases feature. The noun phrases feature (f11) is actually in the top nine features with the highest coefficient but its contribution was not significant at the 0.05 level ($p = 0.068$).

The importance of question focus for *why*-QA is especially interesting because it is a question feature that is specific to *why*-questions and does not similarly apply

Table 2
Features that significantly contribute to the re-ranking score ($p < 0.05$), ranked by their coefficient in the logistic regression model (representing their importance).

Feature	Coefficient
TF-IDF (f0)	0.39**
Overlap between question focus synonyms and document title (f30)	0.25**
Overlap between question object synonyms and answer words (f28)	0.22
Overlap between question object and answer objects (f10)	0.18*
Overlap between question words and document title synonyms (f33)	0.17
Overlap between question verb synonyms and answer words (f24)	0.16
WordNet Relatedness (f35)	0.16*
Cue phrases (f36)	0.15*

Asterisks on coefficients denote the level of significance for the feature: ** $p < 0.001$; * $0.001 < p < 0.01$; no asterisk means $0.01 < p < 0.05$.

to factoids or other question types. Moreover, the link from the question focus to the document title shows that Wikipedia as an answer source can provide QA systems with more information than a collection of plain texts with less discriminative document titles does.

The significance of cue phrases is also an important finding. In fact, including cue phrases in the *why*-QA process is the only feasible way of specifying which passages are likely to contain an explanation (i.e., an answer to a *why*-question). In earlier work (Verberne et al. 2007a), we pointed out that higher-level annotation such as discourse structure can give useful information in the *why*-answer selection process. However, the development of systems that incorporate discourse structure suffers from the lack of tools for automated annotation. The current results show that surface patterns (the literal presence of items from a fixed set of cue words) are a step in the direction of answer selection.

The significant features in Table 2 also show us which question constituents are the most salient for answer ranking: focus, main verb, and direct object. We think that features incorporating the question's subject are not found to be significant because, in a subset of the questions, the subject is semantically poor. Moreover, because for most questions the subject is the question focus, the subject features and the focus features are correlated. In our data, the question focus apparently is the more powerful predictor.

6.2 Comparison to Other Approaches

The 23% improvement that we reach in terms of MRR@150 (from 0.25 to 0.34) is comparable to that reached by Tiedemann in his work on improving factoid QA with the use of structural information.

In order to see whether the improvement that we achieved with re-ranking is on account of structural information or just the benefit of using word sequences, we experimented with a set of re-ranking features based on sequences of question words that are not syntactically defined. In this re-ranking experiment, we included TF-IDF, word bigrams, and word trigrams as features. The resulting performance was around baseline level (MRR = 0.25), significantly worse than re-ranking with structural overlap features. This is still true if we add the cue word feature (which, in isolation, only gives a small improvement to baseline performance) to the *n*-gram features.

6.3 Solving the Remaining Problems

Although the results in terms of success@10 and MRR@150 are satisfactory, there is still a substantial proportion of *why*-questions that is not answered in the top 10 result list. In this section, we discuss a number of attempts that we made to further improve our system.

First, after we found that for some question parts synonym expansion leads to improvement (especially the main verb and direct object), we experimented with the addition of synonyms for these constituents in the retrieval step of our system (Lemur). We found, however, that it does not improve the results due to the large synonym sets of many verbs and nouns which add much noise and lead to very long queries. The same holds for the addition of cue words in the retrieval step.

Second, although our re-ranking module incorporates expansion to synonym sets, there are many question–answer pairs where the vocabulary gap between the question

and the answer is still a problem. There are cases where semantically related terms in the question and the answer are of different word classes (e.g., *hibernate—hibernation*), and there are cases of proper nouns that are not covered by WordNet (e.g., *B.B. King*). We considered using dynamic stemming for verb–noun relations such as the *hibernation* case but research has shown that stemming hurts as many queries as it helps (Bilotti, Katz, and Lin 2004). Therefore, we experimented with a number of different semantic resources, namely, the nominalization dictionary Nomlex (Meyers et al. 1998) and the wikiOntology by Ponzetto and Strube (2007). However, in their current state of development these semantic resources cannot improve our system because their coverage is too low to make a contribution to our re-ranking module. Moreover, the present version of the wikiOntology is very noisy and requires a large amount of cleaning up and filtering.

Third, we considered that the use of cue phrases may not be sophisticated enough for finding explanatory relations between question and answer. Therefore, we experimented with the addition of cause–effect pairs from the English version of the EDR Concept Dictionary (Yokoi 1995) — as suggested by Higashinaka and Isozaki (2008). Unfortunately, the list appeared to be extremely noisy, proving it not useful as a source for answer ranking.

7. Conclusions and Directions for Future Research

In the current research, we extended a passage retrieval system for *why*-QA using off-the-shelf retrieval technology (Lemur/TF-IDF) with a re-ranking step incorporating structural information. We get significantly higher scores in terms of MRR@150 (from 0.25 to 0.34) and success@10. The 23% improvement that we reach in terms of MRR is comparable to that reached on various other QA tasks by other researchers in the field (see Section 6.3). This confirms our hypothesis in Section 1 that for the relatively complex problem of *why*-QA, a significant improvement can be gained by the addition of structural information to the ranking component of the QA system.

Most of the features that we implemented for answer re-ranking are based on word overlap between part of the question and part of the answer. As a result of this set-up, our features identify the parts of *why*-questions and their candidate answers that are the most powerful/effective for ranking the answers. The question constituents that appear to be the most important are the question focus, the main verb, and the direct object. On the answer side, most important are the title of the document in which the candidate answer is embedded and knowledge on the presence of cue phrases.

Because our features are overlap-based, they are relatively easy to implement. For implementation of some of the significant features, a form of syntactic parsing is needed that can identify subject, verb, and direct object from the question and sentences in the candidate answers. An additional set of rules is needed for finding the question focus. Finally, we need a fixed list for identifying cue phrases. Exploiting the title of answer documents in the feature set is only feasible if the documents that may contain the answers have titles and section headings similar to Wikipedia.

In conclusion, we developed a method for significantly improving a BOW-based approach to *why*-QA that can be implemented without extensive semantic knowledge sources. Our series of experiments suggest that we have reached the maximum performance that can be obtained using a knowledge-poor approach. Experiments with more complex types of information (discourse structure, cause–effect relations) show that these information sources have not as yet developed sufficiently to be exploited in a QA system.

References

Bilotti, M. W., B. Katz, and J. Lin. 2004. What works better for question answering: Stemming or morphological query expansion. In *Proceedings of the Workshop on Information Retrieval for Question Answering (IR4QA) at SIGIR 2004*, Sheffield.

Bilotti, M. W., P. Ogilvie, J. Callan, and E. Nyberg. 2007. Structured retrieval for question answering. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 351–358, Amsterdam.

Burnage, G., R. H. Baayen, R. Piepenbrock, and H. van Rijn. 1990. *CELEX: A Guide for Users*. CELEX, University of Nijmegen, the Netherlands.

Carlson, Lynn, Daniel Marcu, and Mary Ellen Okurowski. 2003. Building a discourse-tagged corpus in the framework of Rhetorical Structure Theory. In Jan van Kuppevelt and Ronnie Smith, editors, *Current Directions in Discourse and Dialogue*. Kluwer Academic Publishers, Dordrecht, pages 85–112.

Charniak, E. 2000. A maximum-entropy-inspired parser. *ACM International Conference Proceeding Series*, 4:132–139.

Denoyer, L. and P. Gallinari. 2006. The Wikipedia XML corpus. *ACM SIGIR Forum*, 40(1):64–69.

Ferret, O., B. Grau, M. Hurault-Plantet, G. Illouz, L. Monceaux, I. Robba, and A. Vilnat. 2002. Finding an answer based on the recognition of the question focus. *NIST Special Publication*, pages 362–370.

Higashinaka, R. and H. Isozaki. 2008. Corpus-based question answering for why-questions. In *Proceedings of IJCNLP*, pages 418–425, Hyderabad.

Hovy, E. H., U. Hermjakob, and D. Ravichandran. 2002. A question/answer typology with surface text patterns. In *Proceedings of the Human Language Technology conference (HLT)*, pages 247–251, San Diego, CA.

Joachims, T., L. Granka, B. Pan, H. Hembrooke, and G. Gay. 2005. Accurately interpreting clickthrough data as implicit feedback. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 154–161, Salvador, Brazil.

Khalid, M. and S. Verberne. 2008. Passage retrieval for question answering using Sliding Windows. In *Proceedings of the COLING 2008 Workshop IR4QA*, Manchester, UK.

Liu, T. Y., J. Xu, T. Qin, W. Xiong, and H. Li. 2007. Letor: Benchmark dataset for research on learning to rank for information retrieval. In *Proceedings of the Workshop on Learning to Rank for Information Retrieval (LR4IR) at SIGIR 2007*, pages 3–10, Amsterdam.

Meyers, A., C. Macleod, R. Yangarber, R. Grishman, L. Barrett, and R. Reeves. 1998. Using NOMLEX to produce nominalization patterns for information extraction. In *Proceedings: The Computational Treatment of Nominals*, volume 2, pages 25–32, Montreal.

Pedersen, T., S. Patwardhan, and J. Michelizzi. 2004. WordNet::Similarity — measuring the relatedness of concepts. In *Proceedings of the National Conference on Artificial Intelligence*, pages 1024–1025, San Jose, CA.

Ponzetto, S. P. and M. Strube. 2007. Deriving a large scale taxonomy from Wikipedia. In *Proceedings of the National Conference on Artificial Intelligence*, pages 1440–1445, Vancouver, BC.

Quarteroni, S., A. Moschitti, S. Manandhar, and R. Basili. 2007. Advanced structural representations for question classification and answer re-ranking. In *Proceedings of ECIR 2007*, pages 234–245, Rome.

Salton, G. and C. Buckley. 1988. Term-weighting approaches in automatic text retrieval. *Information Processing and Management*, 24(5):513–523.

Surdeanu, M., M. Ciaramita, and H. Zaragoza. 2008. Learning to rank answers on large online QA collections. In *Proceedings of ACL 2008*, pages 719–727, Columbus, OH.

Tellex, S., B. Katz, J. Lin, A. Fernandes, and G. Marton. 2003. Quantitative evaluation of passage retrieval algorithms for question answering. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 41–47, Toronto.

Tiedemann, J. 2005. Improving passage retrieval in question answering using NLP. In *Progress in Artificial Intelligence*, volume 3808. Springer, Berlin / Heidelberg, pages 634–646.

Verberne, S. 2006. Developing an approach for why-question answering. In *Conference Companion of the 11th Conference of the European Chapter of the Association for*

Computational Linguistics (EACL 2006), pages 39–46, Trento.

Verberne, S., L. Boves, N. Oostdijk, and P. A. Coppen. 2007a. Discourse-based answering of why-questions. *Traitement Automatique des Langues (TAL), special issue on "Discours et document: traitements automatiques"*, 47(2):21–41.

Verberne, S., L. Boves, N. Oostdijk, and P. A. Coppen. 2007b. Evaluating discourse-based answer extraction for why-question answering. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 735–736, Amsterdam.

Verberne, S., H. Van Halteren, D. Theijssen, S. Raaijmakers, and L. Boves. 2009. Learning to rank QA data. In *Proceedings of the Workshop on Learning to Rank for Information Retrieval (LR4IR) at SIGIR 2009*, pages 41–48, Boston, MA.

Yokoi, T. 1995. The EDR electronic dictionary. *Communications of the ACM*, 38(11):42–44.

Zhai, C. 2001. Notes on the Lemur TFIDF model. Technical report, School of Computer Science, Carnegie Mellon University.