Passage Reranking for Question Answering Using Syntactic Structures and Answer Types

Elif Aktolga, James Allan, and David A. Smith

Center for Intelligent Information Retrieval, Department of Computer Science,
University of Massachusetts Amherst, Amherst MA, 01002, USA
{elif,allan,dasmith}@cs.umass.edu

Abstract. Passage Retrieval is a crucial step in question answering systems, one that has been well researched in the past. Due to the vocabulary mismatch problem and independence assumption of bag-of-words retrieval models, correct passages are often ranked lower than other incorrect passages in the retrieved list. Whereas in previous work, passages are reranked only on the basis of syntactic structures of questions and answers, our method achieves a better ranking by aligning the syntactic structures based on the question's answer type and detected named entities in the candidate passage. We compare our technique with strong retrieval and reranking baselines. Experimental results using the TREC QA 1999-2003 datasets show that our method significantly outperforms the baselines over all ranks in terms of the MRR measure.

Keywords: Passage Retrieval, Question Answering, Reranking, Dependency Parsing, Named Entities.

1 Introduction

Prior work in the area of factoid question answering (QA) showed that the passage retrieval phase is critical [2], [10], [11]: many of the retrieved passages are non-relevant because they do not contain a correct answer to the question despite the presence of common terms between the retrieved passages and the question. Therefore such passages must be eliminated post-retrieval. Otherwise, this affects the succeeding answer extraction phase, resulting in incorrect answers. One cause of this problem is a failure to consider dependencies between terms in the original question and matching query terms in the passage during retrieval. This issue was tackled in recent work [2], [11]: a better ranking was obtained by analyzing syntactic and semantic transformations in correct question and answer pairs, utilizing the fact that if a passage contains an answer to a question (we call this a true QA pair), there is some similarity in their parse structures.

The main problem with approaches in previous work is that dependencies between parts of sentences of a QA pair are evaluated without first considering whether the candidate sentence bears an answer to the question at all [2]. Wrong passages could easily be eliminated by scanning them for possible answer candidates, an approach also employed by humans when searching for an

P. Clough et al. (Eds.): ECIR 2011, LNCS 6611, pp. 617–628, 2011.

answer to a question. Since certain syntactic substructures such as noun or verb phrases frequently co-occur in sentences irrespective of the QA pair being true, passages that are merely scored based on such parse structure similarities with the question yield a suboptimal ranking.

For factoid question answering we therefore propose an improved method that performs the candidate-answer check by determining the answer type of the question. For example, 'Who is John?' has the answer type *person* and 'Where is Look Park?' has the answer type *location*. For such questions we know that the answer passage must contain a named entity of the detected answer type, otherwise the passage is likely to be non-relevant. If a passage passes this initial check, then the parse structures of the QA pair can be analyzed with respect to the named entities that are found as candidate answers for determining the relevance of the answer passage. This ensures that only relevant parts of the syntactic structures are compared and evaluated, and that passages are not accidentally ranked highly if they contain some common subphrase structures with the question.

In this paper, we view the task of enhancing passage retrieval from two angles: the first objective is to improve retrieval per se, i.e. to increase the number of questions for which we retrieve at least one passage containing the right answer. For this, we experiment with various passage retrieval models detailed in Section 3. Then, we aim at obtaining a better ranking so that correct passages are ranked higher than incorrect ones. Our results demonstrate that our improved reranking technique performs up to 35.2% better than previous methods when interpolated with the original retrieval scores.

2 Related Work

Question Answering (QA) has been researched extensively since the beginning of the TREC QA evaluations in the late 1990s. Typically, QA systems are organized in a pipelined structure, where the input to the system is a natural language question and the output is a ranked list of n answers. The importance of the retrieval component in a QA system has been highlighted in the field [2], [9], [10], [11]. If a bad set of passages is retrieved, not even containing a single answer to the posed question, the QA system fails at the retrieval step itself for that question. The ranking of the passages is also very important, since answer extraction is typically applied to the top-ranked list of retrieved passages rather than to the whole set.

In order to overcome the limitations of passage retrieval, reranking methods have been applied as a post-retrieval step. Cui et al. [2] proposed a fuzzy relation matching technique that learns a dependency parse relation mapping model by means of expectation maximization (EM) and translation models. This model measures sentence structure similarity by comparing dependency label paths between matched terms in questions and answers. When matching terms, only the most probable alignment of terms is considered. Wang et al. [11] developed an improved approach to reranking passages: their model learns soft alignments

by means of EM, however instead of translation models they used a probabilistic quasi-synchronous grammar that allows more flexible matching of subsets of parse trees (called 'configurations') rather than single terms. Quasi-synchronous grammars were originally proposed by Smith and Eisner [8] for machine translation. Unlike Cui et al.'s approach, the final score for a passage under this approach is obtained by summing over all alignments of terms. Our work combines some of the advantages from both papers, extending them in a new direction: we train Cui et al.'s model by means of EM and translation models *with respect to the question's detected answer type and named entities in the answer passage.* We also employ more flexible matching of terms by means of Wordnet synonyms and we sum over all alignment scores as Wang et al.

We use the open domain question answering system OpenEphyra 0.1.1 as our system [5], [6], [7], which is a full implementation of a pipelined question answering system. Since we only measure passage retrieval and reranking performance, we disabled the answer extraction component. For analyzing parse structures of questions and answers, we integrated the dependency parser MSTParser [3] into the system, and extended OpenEphyra further for our passage retrieval and reranking algorithms.

3 Passage Retrieval

In this section we describe the passage retrieval techniques that are applied before passage reranking. All our algorithms employ the query likelihood language model as their basis. The differences in our techniques are in how query generation and formulation are achieved.

We create our passages as follows: paragraphs in the TREC QA datasets are processed with OpenNLP's sentence detector so that they can be broken down into sentences where possible. So ideally passages correspond to sentences. This yields the best results in our experiments, since returned passages contain as little non-relevant material as possible. Since we only apply our methods to factoid questions, for which the answer is always contained within a single sentence, this representation is sufficient. Further, this allows us to compare our methods to previous work, such as Cui et al. [2].

3.1 Bag Of Words (Q-BOW)

We begin with our simplest baseline model *Q-BOW*, which is just a query likelihood of unigram phrases. Mathematically, we can state this model as:

$$P(Q|D) = P(q_1, \cdots, q_n|M_D) = \prod_{i=1}^{n} P(q_i|M_D) \tag{1}$$

where M_D is the language model of passage D. We use Dirichlet smoothing with $\mu = 2500$ for our experiments.

3.2 Adding Question Analysis (QuAn)

For the next baseline, *QuAn*, we extend *Q-BOW* in several ways: we allow the addition of n-gram phrases as keywords so that the model does not consist of unigrams only; further, we add the output of OpenEphyra's front end question analysis phase. This generates two types of phrase queries from a question, question reformulations and question interpretations [5].

Mathematically, we use the same model as in (1), with the difference that the q_i can now be n-gram phrases referring to reformulations, interpretations, or phrases extracted from the question.

3.3 Expanding with Synonyms (QuAn-Wnet)

QuAn-Wnet is a further extension of our previous techniques, for which we expand the query generated by *QuAn* with at most 10 keywords obtained from Wordnet. These keywords are n-gram synonyms of phrases generated through *Q-BOW* and *QuAn*.

4 Passage Reranking

The next step is to rerank passages, for which we take the output from the passage retrieval phase as generated by *QuAn-Wnet*, and apply one of our reranking techniques to it. We use *QuAn-Wnet* since it performs best (Section 5).

4.1 Extraction of Dependency Relation Paths

A dependency parse of a sentence yields a parse tree consisting of nodes that correspond to words in the sentence and labelled edges describing the type of *dependency relation* between two connected nodes or words. Therefore, a *dependency relation path* is a sequence of dependency relations between two nodes in the parse tree. For example, from the parse of the sentence 'But there were no buyers' depicted in Figure 1, we can infer the relation path `DEP NP-PRD DEP` between the words 'But' and 'no'. Dependency relations are usually directed, but we ignore the directions of dependency relations for the analysis of questions and answers as in previous work [2].

Let q_i denote a word in the question $Q = q_1, \cdots, q_n$ and a_i a word in the answer $A = a_1, \cdots, a_m$. We extract relation paths from questions and answers

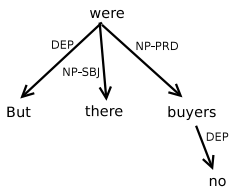

Fig. 1. Dependency Parse tree for the sentence 'But there were no buyers'

whenever we find a matching pair $[(q_k, a_l), (q_r, a_s)]$, where $k \neq r$ and $l \neq s$. In this work, a *match* between two words w_i and w_j occurs if w_i and w_j share the same root (or stem) or if w_i is a synonym of w_j. Therefore matching words can only be nouns, verbs, adjectives, and adverbs. For our improved reranking methods employing answer type checking and named entities, we also require matches between question words and candidate answer terms (see Section 4.4). As is the case with IBM translation models, words are matched in a many-to-one fashion due to computational reasons. That is, each word in the question is assigned at most one word from the answer, but answer words can be assigned to multiple question words.

In order to decide whether an answer passage and a question belong together, we compare all possible extracted dependency relation paths from the question and the answer. The idea behind this approach is to find certain patterns of relation paths by means of which we can detect true QA pairs to score them higher than other pairs. This is the foundation of the dependency relation path based reranking methods detailed below.

4.2 Training the Model

Given a matching pair $[(q_k, a_l), (q_r, a_s)]$, we extract from this the relation paths $\langle path_q, path_a \rangle$, where $path_q$ is the relation path between q_k and q_r in the question and $path_a$ is the path between a_l and a_s in the answer accordingly. In order to compare the extracted dependency relation paths $\langle path_q, path_a \rangle$, we need a model that captures which relation paths are more probable to be seen together in true QA pairs. For this, we trained a translation model with GIZA++ using IBM Model 1 as described by Cui et al. [2]. We extracted 14009 true QA pairs from sentences in our training set of the TREC QA 1999-2003 corpora. The trained translation model on the relation paths $\langle path_q, path_a \rangle$ then yields the probability $P(label_a|label_q)$, where $label_a$ is a single dependency relation in $path_a$, and $label_q$ is a relation in $path_q$. We use these probabilities in our reranking techniques detailed below to score $\langle path_q, path_a \rangle$.

4.3 Dependency Relation Path Similarity (Cui)

This technique is our improved implementation of Cui et al.'s [2] approach. It reranks passages only based on the similarity of the dependency relation paths between passages and questions. More specifically, given a question Q and an answer candidate passage A, we score A as follows:

$$\sum_{\langle path_q, path_a \rangle \in Paths} scorePair(path_q, path_a) \tag{2}$$

That is, we sum up the scores of all extracted $\langle path_q, path_a \rangle$ pairs by aligning Q and A in all possible ways [11]. This is different from previous work [2], where only the most probable alignment is considered. The score for an individual path pair $scorePair(path_q, path_a)$ is then calculated as follows:

$$\frac{1}{|path_a|} \prod_{label_{a_j}} \sum_{label_{q_i}} P(label_{a_j}|label_{q_i}) \tag{3}$$

where $P(label_{a_j}|label_{q_i})$ is obtained from our trained translation model, and $\frac{1}{|path_a|}$ is used as a normalization factor, since the lengths of the answer paths vary whereas those of the question remain the same [2]. We adapt an IBM Model 1 way of scoring here with a product of a sum over the labels since we consider all possible alignments. The derivation of this formula for translation modeling is described in detail in Brown et al.'s work [1].

4.4 Dependency Relation Path Similarity with Answer Type Checking and Named Entities (Atype-DP)

This is our improved reranking method for which we employ answer type checking and the analysis of named entities (NE) in the candidate answer passages. For answer type checking, we use OpenEphyra's answer type classification module [7], which can detect 154 answer types organized in different hierarchies. Therefore, often several answer types of varying granularity are assigned to a question. This allows a greater flexibility when matching answer types to named entities. The detected answer types in the question are used to look for candidate answer terms in passages. For named entity detection, we use OpenEphyra's built-in named entity extraction module [5], which comprises about 70 NE types.

For this model, we revise the definition of how *matching* is performed, originally introduced in Section 4.1: a matching pair is now a tuple $[(qword, aCand), (q_r, a_s)]$, where $qword \neq q_r$ and $aCand \neq a_s$, with $qword$ being the question word (e.g. 'who'), and $aCand$ a candidate answer term (e.g. 'Kate') matching the question word's answer type. (q_r, a_s) are other words fulfilling the earlier criteria for matching (words sharing the same root or being synonyms). Figure 2 shows an example.

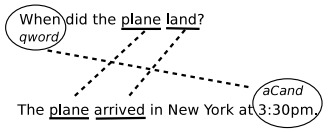

Fig. 2. QA pair with the question word, candidate answer, and matching terms highlighted. There are two matching pairs: [(when,3:30pm),(plane,plane)] and [(when,3:30pm),(land,arrived)].

Note that there can be multiple answer candidates of the same NE type in a single answer passage. In this model, we consider all alignments of Q and A *given the best answer candidate*. Under this model, the score for a passage A can formally be stated as:

$$\max_i \sum_{\langle path_q, path_a \rangle \in Paths_{aCand_i}} scorePair(path_q, path_a) \tag{4}$$

where $Paths_{aCand_i}$ are all dependency relation paths $\langle path_q, path_a \rangle$ extracted from matching pairs containing the answer candidate $aCand_i$. The calculation of score $scorePair(path_q, path_a)$ remains the same as (3), with the exception that we use a retrained translation model with our revised matching approach. The probabilities $P(label_{a_j}|label_{q_i})$ are therefore obtained from this new translation model.

The main difference of this reranking approach to previous techniques [2], [11] lies in how we perform matching: by considering the answer type and the best answer candidate during matching, we ensure that only relevant dependency relation paths between the candidate answer (or question word) and other matching terms are considered. This way, the obtained score reflects the dependencies within a passage towards its relevance for a given question more accurately. In previous approaches, arbitrary relation paths were compared in QA pairs, which often does not yield a good ranking of passages.

4.5 Interpolation with Retrieval Baseline (Atype-DP-IP)

This is a variation of the *Atype-DP* reranking method where we score a passage A given a question Q by interpolating *Atype-DP* with the best retrieval baseline *QuAn-Wnet* as follows:

$$score(A) = \lambda score_{\text{QuAn-Wnet}}(A) + (1 - \lambda) score_{\text{Atype-DP}}(A) \tag{5}$$

$score_{\text{QuAn-Wnet}}$ is the retrieval score, whereas $score_{\text{Atype-DP}}$ is our reranking score. In order to interpolate the scores we normalize them so that they are between 0 and 1 and rerank passages based on this new score. The advantage of this approach is that the original retrieval scores are not discarded and impact the results. This helps in cases where *Atype-DP* does not work well due to poor named entity detection, as we will see in Section 5.

4.6 Elimination of Non-Answer-Type-Bearing Passages (QuAn-Elim)

This approach is different from the other reranking methods in that it does not rearrange the order of the retrieved passages, but it eliminates retrieved passages that are likely to not contain an answer based on candidate named entity matches, acting similarly to a passage answer type filter in other QA systems [4]. This approach is interesting for comparing with our reranking methods *Atype-DP* and *Atype-DP-IP* that involve an analysis of parse structures.

5 Experiments

The objectives of our research are (1) to show that we can improve the retrieval of passages by employing models that exploit findings from the question analysis phase and synonyms of query terms; (2) to demonstrate that we can obtain a better ranking of passages by employing a candidate-answer check and by

Table 1. Evaluation of Retrieval Baselines. All results are averages from the testing datasets TREC 2000 and TREC 2001.

Model	*Success@5*	*Success@10*	*Success@20*	*Success@100*
Q-BOW	0.325	0.43	0.512	0.732
QuAn	0.356	0.457	0.552	0.751
QuAn-Wnet	**0.364***	**0.463**	**0.558**	**0.753**

analyzing their parse structure with respect to candidate named entities. We compare our reranking techniques *Atype-DP* and *Atype-DP-IP* with *Cui* and *QuAn-Elim* only, since in Cui et al.'s paper the technique was shown to be superior to various existing methods [2].

5.1 Data and Evaluation

For all our experiments, we use the TREC QA 1999-2003 corpora, questions, and evaluation data. For correct evaluation, we eliminated NIL questions. We split the questions into a test set with 1125 questions, consisting of the TREC QA 2000 and TREC QA 2001 data, and a training set, comprising of the remaining 1038 TREC QA 1999-2003 questions. The training set is also used for the translation models described in Section 4.2.

We determine the correctness of a passage with respect to the question as follows: we consider a passage as relevant and bearing the correct answer to a question if (1) it comes from a document with a correct document ID, and (2) it contains at least one answer pattern.

In the next sections, we report the results for the retrieval and reranking performances separately. For measuring retrieval performance, we use the success measure, which determines the percentage of correctly answered questions at different ranks. For reporting passage ranking quality, we utilize the mean reciprocal rank measure (MRR), which has widely been used in QA for answer ranking.

5.2 Retrieval Performance

For this evaluation we utilized 1074 questions from TREC 2000 and TREC 2001 – these are all questions for which queries for retrieval were successfully generated by the question analysis phase in OpenEphyra. Table 1 shows the results of the runs with the *Q-BOW*, *QuAn*, and *QuAn-Wnet* baselines. We can clearly see that over all ranks, as the baseline retrieval method uses more sophisticated queries, the percentage of correctly answered questions increases in terms of the success measure. Since *QuAn-Wnet* performs best (and significantly better with p-value smaller than 0.01 for high ranks), we used this retrieval baseline for our further experiments.

5.3 Reranking Performance

For our reranking evaluation, we utilized 622 questions from TREC 2000 and TREC 2001 due to the following requirements: (1) Successful question analysis

Table 2. Evaluation of Reranking Techniques. All results are averages from the testing datasets TREC 2000 and TREC 2001, evaluated on the top 100 retrieved passages.

Model	*MRR@1*	*MRR@5*	*MRR@10*	*MRR@20*	*MRR@50*	*MRR@100*
Q-BOW	0.168	0.266	0.286	0.293	0.299	0.301
QuAn-Wnet	0.193	0.289	0.308	0.319	0.324	0.325
Cui	0.202	0.307	0.325	0.335	0.339	0.341
Atype-DP	0.148	0.24	0.26	0.273	0.279	0.28
Atype-DP-IP	**0.261***	**0.363***	**0.38***	**0.389***	**0.393***	**0.394***
% Improvement over Cui	**+29.2**	+18.24	+16.9	+16.12	+15.9	+15.54
% Improvement over QuAn-Wnet	**+35.2**	+25.6	+23.4	+21.9	+21.3	+ 21.2

is required as for retrieval: We eliminate questions for which OpenEphyra could not generate a query; (2) As with retrieval, at least one true passage must be present within the first 100 ranks of retrieved passages with *QuAn-Wnet*, since this is the range of passages we parse and analyze. (3) Candidate answer type detection with OpenEphyra must be successful. Since we measure the success of our reranking techniques depending on candidate answer type checking, we only evaluate those questions for which our system detects an answer type.

Table 2 illustrates the reranking results with the techniques *Cui*, *Atype-DP*, *Atype-DP-IP*, and the retrieval baselines. We first note that while our reranking approach *Atype-DP* performs worst, its interpolated version *Atype-DP-IP* significantly outperforms all other techniques with p-value less than 10^{-4} over all ranks. The highest percent improvement is at rank 1 with a gain of 29.2% over *Cui* and 35.2% over the retrieval baseline *QuAn-Wnet*. We investigated the reasons for this performance: *Atype-DP* depends on successful named entity extraction, since dependency relation paths are extracted from passages between matching terms and a candidate answer of the required answer type. For many passages though, OpenEphyra's named entity extraction module could not detect a single named entity of the required answer type. Hence, affected passages are ranked low.

We further observe that our enhanced implementation of Cui et al.'s technique *Cui* [2] performs worse than *Atype-DP-IP* and only a little better than *QuAn-Wnet*. Wang et al. [11] also report the method being rather brittle. This technique does not depend on other factors than retrieval and dependency parsing. Figure 3 provides a more accurate view of the results: The MRR scores of *Cui* and *Atype-DP-IP* are plotted as a cumulative distribution function. Over all ranks, *Atype-DP-IP* has a smaller fraction of questions with smaller MRRs than *Cui*.

We found that the technique *QuAn-Elim*, which does not use any parse structure information – but merely eliminates non-answer bearing passages – on average performs comparably to *Atype-DP-IP*, although it slightly suffers at rank 1 with an MRR of 0.243. This difference is however not significant. A detailed analysis of the three methods *Atype-DP*, *Atype-DP-IP*, and *QuAn-Elim* can be seen in Figure 4: we compared the differences in ranking of the first correct answer

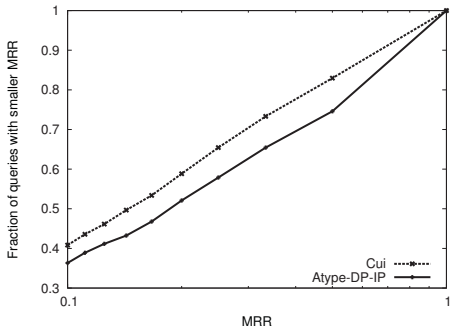

Fig. 3. Cumulative distribution function of the MRR scores in logscale (lower is better)

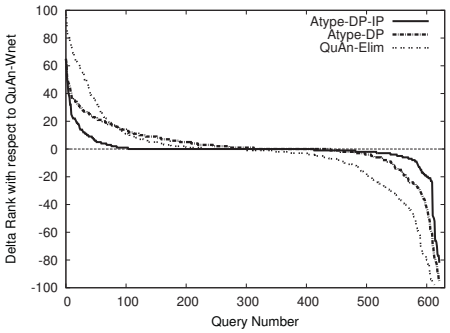

Fig. 4. Differences in Ranking of the first correct answer passage in Atype-DP-IP, Atype-DP, and QuAn-Elim with respect to the retrieval baseline QuAn-Wnet

passage of these methods with respect to the baseline *QuAn-Wnet*. *Atype-DP* and *Atype-DP-IP* improve about 310 questions, whereas *QuAn-Elim* only ranks 230 questions higher than the baseline. Note that *QuAn-Elim* ranks some of these questions higher than the other methods. *Atype-DP* also ranks questions higher than *Atype-DP-IP*, since the parse structure analysis with respect to NEs works well. In the middle range, *Atype-DP-IP* has a larger number of questions whose first correct answer passage appears in the same rank as in the retrieval baseline. We analyzed these 300 questions, of which 119 already have a correct answer at rank 1, so they cannot be improved further. On the right end we can observe that *QuAn-Elim* decreases the ranking of correct answers for more questions, and to a higher degree than the other two methods: whereas we get a good gain for a small number of questions, the method worsens performance for about 260 questions. As for the other methods we notice that *Atype-DP* ranks roughly the same amount of questions much lower than *Atype-DP-IP*: This is where the interpolation helps. If named entity detection does not work, the retrieval score prevents the passage from being ranked low. Increasing the impact of the retrieval score further however disturbs the ranking as it can be seen in Figure 5: we adjusted the interpolation parameter λ to achieve the best performance at $\lambda = 0.7$ as observed in training data. These observations support the hypothesis that the reranking technique is useful on its own, and that the retrieval score aids in recovering errors arising from poor named entity detection.

The results in Table 2 suggest that since *Atype* performs better than all other techniques by means of a little tweak with the baseline, then *Cui* should perform even better than *Atype-DP-IP* when interpolated with the retrieval baseline. Surprisingly, the results in Table 3 and Figure 5 show that *Cui* does not improve with the same effect. This supports our hypothesis that reranking by parse structure analysis with respect to candidate named entities is more effective in optimizing precision.

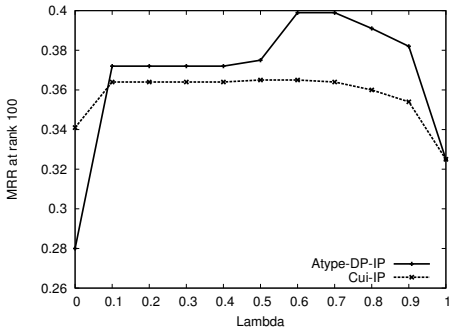

Fig. 5. Effect of varying λ in the interpolated methods Atype-DP-IP and Cui-IP

Table 3. Average MRR results of interpolating Atype-DP-IP and Cui-IP with $\lambda = 0.7$ for the retrieval baseline

Model	*MRR@1*	*MRR@5*	*MRR@10*	*MRR@20*	*MRR@50*	*MRR@100*
Atype-DP-IP	0.261	0.363	0.38	0.389	0.393	0.394
Cui-IP	0.225	0.331	0.347	0.356	0.361	0.362

6 Conclusions

We have presented a passage reranking technique for question answering that utilizes parse structures of questions and answers in a novel manner: the syntactic structures of answer passages are analyzed with respect to present candidate answer terms, whose type is determined by the question's answer type. This way we ensure that only relevant dependency relation paths between the candidate answer and other parts of the sentence are analyzed with respect to the corresponding paths in the question. Our results show that this method outperforms previous approaches and retrieval baselines up to 35.2%, given that cases where named entity extraction does not succeed are backed off to the retrieval score. In future work, we expect that utilizing a more accurate named entity extraction module with respect to answer types will improve the results even further. It would also be interesting to compare our reranking techniques with those of Wang et al., who take a completely different approach to the problem.

Acknowledgments. This work was supported in part by the Center for Intelligent Information Retrieval. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors' and do not necessarily reflect those of the sponsor.

References

1. Brown, P.F., Pietra, V.J., Pietra, S.A.D., Mercer, R.L.: The mathematics of statistical machine translation: Parameter estimation. Computational Linguistics 19, 263–311 (1993)

2. Cui, H., Sun, R., Li, K., Kan, M.-Y., Chua, T.-S.: Question answering passage retrieval using dependency relations. In: SIGIR 2005, pp. 400–407. ACM, New York (2005)
3. McDonald, R., Crammer, K., Pereira, F.: Online large-margin training of dependency parsers. In: ACL 2005, Morristown, NJ, USA, pp. 91–98 (2005)
4. Prager, J., Brown, E., Coden, A., Radev, D.: Question-answering by predictive annotation. In: SIGIR 2000, pp. 184–191. ACM, New York (2000)
5. Schlaefer, N., Gieselman, P., Sautter, G.: The Ephyra QA system at TREC 2006 (2006)
6. Schlaefer, N., Gieselmann, P., Schaaf, T., Waibel, A.: A pattern learning approach to question answering within the ephyra framework. In: Sojka, P., Kopeček, I., Pala, K. (eds.) TSD 2006. LNCS (LNAI), vol. 4188, pp. 687–694. Springer, Heidelberg (2006)
7. Schlaefer, N., Ko, J., Betteridge, J., Pathak, M.A., Nyberg, E., Sautter, G.: Semantic Extensions of the Ephyra QA System for TREC 2007. In: TREC (2007)
8. Smith, D.A., Eisner, J.: Quasi-synchronous grammars: Alignment by soft projection of syntactic dependencies. In: Proc. HLT-NAACL Workshop on Statistical Machine Translation, pp. 23–30 (2006)
9. Sun, R., Ong, C.-H., Chua, T.-S.: Mining dependency relations for query expansion in passage retrieval. In: SIGIR 2006, pp. 382–389. ACM, New York (2006)
10. Tellex, S., Katz, B., Lin, J., Fernandes, A., Marton, G.: Quantitative evaluation of passage retrieval algorithms for question answering. In: SIGIR 2003, pp. 41–47. ACM, New York (2003)
11. Wang, M., Smith, N.A., Mitamura, T.: What is the Jeopardy model? A quasi-synchronous grammar for QA. In: EMNLP-CoNLL 2007, pp. 22–32. Association for Computational Linguistics, Prague (2007)