Learning the Relative Usefulness of Questions in Community QA

Razvan Bunescu
School of EECS
Ohio University
Athens, OH 43201, USA
bunescu@ohio.edu

Yunfeng Huang
School of EECS
Ohio University
Athens, OH 43201, USA
yh324906@ohio.edu

Abstract

We present a machine learning approach for the task of ranking previously answered questions in a question repository with respect to their relevance to a new, unanswered reference question. The ranking model is trained on a collection of question groups manually annotated with a partial order relation reflecting the relative utility of questions inside each group. Based on a set of meaning and structure aware features, the new ranking model is able to substantially outperform more straightforward, unsupervised similarity measures.

1 Introduction

Open domain Question Answering (QA) is one of the most complex and challenging tasks in natural language processing. In general, a question answering system may need to integrate knowledge coming from a wide variety of linguistic processing tasks such as syntactic parsing, semantic role labeling, named entity recognition, and anaphora resolution (Prager, 2006). State of the art implementations of these linguistic analysis tasks are still limited in their performance, with errors that compound and propagate into the final performance of the QA system (Moldovan et al., 2002). Consequently, the performance of open domain QA systems has yet to arrive at a level at which it would become a feasible alternative to the current paradigms for information access based on keyword searches.

Recently, community-driven QA sites such as Yahoo! Answers and WikiAnswers [1] have established a new approach to question answering that shifts the inherent complexity of open domain QA from the computer system to volunteer contributors. The computer is no longer required to perform a deep linguistic analysis of questions and generate corresponding answers, and instead acts as a mediator between users submitting questions and volunteers providing the answers.

An important objective in community QA is to minimize the time elapsed between the submission of questions by users and the subsequent posting of answers by volunteer contributors. One useful strategy for minimizing the response latency is to search the QA repository for similar questions that have already been answered, and provide the corresponding ranked list of answers, if such a question is found. The success of this approach depends on the definition and implementation of the question-to-question similarity function. In the simplest solution, the system searches for previously answered questions based on exact string matching with the reference question. Alternatively, sites such as WikiAnswers allow the users to mark questions they think are rephrasings ("alternate wordings", or paraphrases) of existing questions. These question clusters are then taken into account when performing exact string matching, therefore increasing the likelihood of finding previously answered questions that are semantically equivalent to the reference question.

In order to lessen the amount of work required from the contributors, an alternative approach is to build a system that automatically finds rephrasings of questions, especially since question rephrasing

[1] answers.yahoo.com, wiki.answers.com

Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, pages 97–107,
MIT, Massachusetts, USA, 9-11 October 2010.

seems to be computationally less demanding than question answering. According to previous work in this domain, a question is considered a rephrasing of a reference question Q_0 if it uses an alternate wording to express an identical information need. For example, Q_0 and Q_1 below are rephrasings of each other, and consequently they are expected to have the same answer.

Q_0 What should I feed my turtle?

Q_1 What do I feed my pet turtle?

Paraphrasings of a new question cannot always be found in the community QA repository. We believe that computing a ranked list of existing questions that at least partially address the original information need could also be useful to the user, at least until other users volunteer to give an exact answer to the original, unanswered reference question. For example, in the absence of any additional information about the reference question Q_0, the expected answers to questions Q_2 and Q_3 below may be seen as partially overlapping in information content with the expected answer for the reference question Q_0. An answer to question Q_4, on the other hand, is less likely to benefit the user, even though it has a significant lexical overlap with the reference question.

Q_2 What kind of fish should I feed my turtle?

Q_3 What do you feed a turtle that is the size of a quarter?

Q_4 What kind of food should I feed a turtle dove?

In this paper, we propose a supervised learning approach to the question ranking problem, a generalization of the question paraphrasing problem in which questions are ranked in a partial order based on the relative information overlap between their expected answers and the expected answer of the reference question. Underlying the question ranking task is the expectation that the user who submits a reference question will find the answers of the highly ranked questions to be more useful than the answers associated with the lower ranked questions. For the reference question Q_0 above, the learned ranking model is expected to produce a partial order in which Q_1 is ranked higher than Q_2, Q_3 and Q_4, whereas Q_2 and Q_3 are ranked higher than Q_4.

2 Partially Ordered Datasets for Question Ranking

In order to enable the evaluation of question ranking approaches, we have previously created a dataset of 60 groups of questions (Bunescu and Huang, 2010b). Each group consists of a reference question (e.g. Q_0 above) that is associated with a partially ordered set of questions (e.g. Q_1 to Q_4 above). For each reference questions, its corresponding partially ordered set is created from questions in Yahoo! Answers and other online repositories that have a high cosine similarity with the reference question. Out of the 26 top categories in Yahoo! Answers, the 60 reference questions span a diverse set of categories. Figure 1 lists the 20 categories covered, where each category is shown with the number of corresponding reference questions between parentheses.

```
Travel (10), Computers & Internet (6),
Beauty & Style (5), Entertainment &
Music (5), Food & Drink (5), Health (5),
Arts & Humanities (3), Cars &
Transportation (3), Consumer Electronics
(3), Pets (3), Family & Relationships
(2), Science & Mathematics (2),
Education & Reference (1), Environment
(1), Local Businesses (1), Pregnancy &
Parenting (1), Society & Culture (1),
Sports (1), Yahoo! Products (1)
```

Figure 1: The 20 categories represented in the dataset.

Inside each group, the questions are manually annotated with a partial order relation, according to their utility with respect to the reference question. We use the notation $\langle Q_i \succ Q_j | Q_r \rangle$ to encode the fact that question Q_i is *more useful than* question Q_j with respect to the reference question Q_r. Similarly, $\langle Q_i = Q_j \rangle$ will be used to express the fact that questions Q_i and Q_j are reformulations of each other (the reformulation relation is independent of the reference question). The partial ordering among the questions Q_0 to Q_4 above can therefore be expressed concisely as follows: $\langle Q_0 = Q_1 \rangle$, $\langle Q_1 \succ Q_2 | Q_0 \rangle$, $\langle Q_1 \succ Q_3 | Q_0 \rangle$, $\langle Q_2 \succ Q_4 | Q_0 \rangle$, $\langle Q_3 \succ Q_4 | Q_0 \rangle$. Note that we do not explicitly annotate the relation $\langle Q_1 \succ Q_4 | Q_0 \rangle$, since it can be inferred based on the transitivity of the *more useful than* relation: $\langle Q_1 \succ Q_2 | Q_0 \rangle \wedge \langle Q_2 \succ Q_4 | Q_0 \rangle \Rightarrow \langle Q_1 \succ Q_4 | Q_0 \rangle$.

Reference question (Q_r)
Q_5 What's a nice summer camp to go to in Florida?
Paraphrasing questions ($\mathcal{P}$)
Q_6 What camps are good for a vacation during the summer in FL?
Q_7 What summer camps in FL do you recommend?
Useful questions ($\mathcal{U}$)
Q_8 Does anyone know a good art summer camp to go to in FL?
Q_9 Are there any good artsy camps for girls in FL?
Q_{10} What are some summer camps for like singing in Florida?
Q_{11} What is a good cooking summer camp in FL?
Q_{12} Do you know of any summer camps in Tampa, FL?
Q_{13} What is a good summer camp in Sarasota FL for a 12 year old?
Q_{14} Can you please help me find a surfing summer camp for beginners in Treasure Coast, FL?
Q_{15} Are there any acting summer camps and/or workshops in the Orlando, FL area?
Q_{16} Does anyone know any volleyball camps in Miramar, FL?
Q_{17} Does anyone know about any cool science camps in Miami?
Q_{18} What's a good summer camp you've ever been to?
Neutral questions ($\mathcal{N}$)
Q_{19} What's a good summer camp in Canada?
Q_{20} What's the summer like in Florida?

Table 1: A question group.

Also note that no relation is specified between Q_2 and Q_3, and similarly no relation can be inferred between these two questions. This reflects our belief that, in the absence of any additional information regarding the user or the "turtle" referenced in Q_0, we cannot compare questions Q_2 and Q_3 in terms of their usefulness with respect to Q_0.

Table 1 shows another reference question Q_5 from our dataset, together with its annotated group of questions Q_6 to Q_{20}. In order to make the annotation process easier and reproducible, we have divided it into two levels of annotation. During the first annotation stage, each question group is partitioned manually into 3 subgroups of questions:

- $\mathcal{P}$ is the set of *paraphrasing* questions.
- $\mathcal{U}$ is the set of *useful* questions.
- $\mathcal{N}$ is the set of *neutral* questions.

A question is deemed useful if its expected answer may overlap in information content with the expected answer of the reference question. The expected answer of a neutral question, on the other hand, should be irrelevant with respect to the reference question. Let Q_r be the reference question, $Q_p \in \mathcal{P}$ a paraphrasing question, $Q_u \in \mathcal{U}$ a useful question, and $Q_n \in \mathcal{N}$ a neutral question. Then the following relations are assumed to hold among these questions:

1. $\langle Q_p \succ Q_u | Q_r \rangle$: a *paraphrasing* question is more useful than a *useful* question.
2. $\langle Q_u \succ Q_n | Q_r \rangle$: a *useful* question is more useful than a *neutral* question.

Note that as long as these relations hold between the 3 types of questions, the names of the subgroups and their definitions are irrelevant with respect to the implied set of *more useful than* relations, since only the implied ternary relations will be used for training and evaluating question ranking approaches. We also assume that, by transitivity, the following ternary relations also hold: $\langle Q_p \succ Q_n | Q_r \rangle$, i.e. a *paraphrasing* question is more useful than a *neutral* question. Furthermore, if $Q_{p_1}, Q_{p_2} \in \mathcal{P}$ are two paraphrasing questions, this implies $\langle Q_{p_1} = Q_{p_2} | Q_r \rangle$.

For the vast majority of questions, the first annotation stage is straightforward and non-controversial. In the second annotation stage, we perform a finer annotation of relations between questions in the middle group $\mathcal{U}$. Table 1 shows two such relations (using indentation): $\langle Q_8 \succ Q_9 | Q_5 \rangle$ and $\langle Q_8 \succ Q_{10} | Q_5 \rangle$. Question Q_8 would have been a rephrasing of the reference question, were it not for the noun "art" modifying the focus noun phrase "summer camp". Therefore, the information content of the answer to Q_8 is strictly subsumed in the information content associated with the answer to Q_5. Similarly, in Q_9 the focus noun phrase is further specialized through the prepositional phrase "for girls". Therefore, (an answer to) Q_9 is less *useful* to Q_5 than (an answer to) Q_8, i.e. $\langle Q_8 \succ Q_9 | Q_5 \rangle$. Furthermore, the focus "art summer camp" in Q_8 conceptually subsumes the focus "summer camps for singing" in Q_{10}, therefore $\langle Q_8 \succ Q_{10} | Q_5 \rangle$.

We call this dataset *simple* since most of the reference questions are shorter than the other questions in their group. We have also created a *complex* version of the same dataset, by selecting as the reference question in each group a longer question from the same group. For example, if Q_0 were a reference question, it would be replaced with a more complex question, such as Q_2, or Q_3. The annotation is redone to reflect the relative usefulness relations with respect to the new reference questions. We believe that the new *complex* dataset is closer to the actual distribution of questions in community QA repositories: unanswered questions tend to be more specific (longer), whereas general questions (shorter) are more likely to have been answered already. Each dataset is annotated by two annotators, leading to a total of 4 datasets: Simple_1, Simple_2, Complex_1, and Complex_2.

Table 2 presents the following statistics on the two types of datasets (Simple, Complex) for each annotator (1, 2): the total number of paraphrasings ($\mathcal{P}$), the total number of useful questions ($\mathcal{U}$), the total number of neutral questions ($\mathcal{N}$), the total number of *more useful than* ordered pairs encoded in the dataset, either explicitly or through transitivity, and the Inter-Annotator Agreement (ITA). We compute the ITA as the *precision* (*P*) and *recall* (*R*) with respect to the *more useful than* ordered pairs encoded in one annotation ($Pairs_1$) relative to the ordered

Dataset	$\mathcal{P}$	$\mathcal{U}$	$\mathcal{N}$	Pairs	ITA
Simple_1	164	775	594	11015	*P*: 76.6
Simple_2	134	778	621	10436	*R*: 81.6
Complex_1	103	766	664	10654	*P*: 71.3
Complex_2	89	730	714	9979	*R*: 81.3

Table 2: Dataset statistics.

pairs encoded in the other annotation ($Pairs_2$).

$$P = \frac{|Pairs_1 \cap Pairs_2|}{Pairs_1} \quad R = \frac{|Pairs_1 \cap Pairs_2|}{Pairs_2}$$

The statistics in Table 2 indicate that the second annotator was in general more conservative in tagging questions as paraphrases or useful questions.

3 Unsupervised Methods for Question Ranking

An ideal question ranking method would take an arbitrary triplet of questions Q_r, Q_i and Q_j as input, and output an ordering between Q_i and Q_j with respect to the reference question Q_r, i.e. one of $\langle Q_i \succ Q_j | Q_r \rangle$, $\langle Q_i = Q_j | Q_r \rangle$, or $\langle Q_j \succ Q_i | Q_r \rangle$. One approach is to design a *usefulness* function $u(Q_i, Q_r)$ that measures how useful question Q_i is for the reference question Q_r, and define the *more useful than* ($\succ$) relation as follows:

$$\langle Q_i \succ Q_j | Q_r \rangle \Leftrightarrow u(Q_i, Q_r) > u(Q_j, Q_r)$$

If we define $I(Q)$ to be the *information need* associated with question Q, then $u(Q_i, Q_r)$ could be defined as a measure of the relative overlap between $I(Q_i)$ and $I(Q_r)$. Unfortunately, the information need is a concept that, in general, is defined only intensionally and therefore it is difficult to measure. For lack of an operational definition of the information need, we will approximate $u(Q_i, Q_r)$ directly as a measure of the similarity between Q_i and Q_r. The similarity between two questions can be seen as a special case of text-to-text similarity, consequently one possibility is to use a general text-to-text similarity function such as *cosine similarity* in the vector space model (Baeza-Yates and Ribeiro-Neto, 1999):

$$cos(Q_i, Q_r) = \frac{Q_i^T Q_r}{\|Q_i\| \|Q_r\|}$$

Here, Q_i and Q_r denote the corresponding *tf*$\times$*idf* vectors.

As a measure of question similarity, one major drawback of cosine similarity is that it is oblivious of the meanings of words in each question. This particular problem is illustrated by the three questions below. Q_{22} and Q_{23} have the same cosine similarity with Q_{21}, they are therefore indistinguishable in terms of their usefulness to the reference question Q_{21}, even though we expect Q_{22} to be more useful than Q_{23} (a place that sells hydrangea often sells other types of plants too, possibly including cacti).

Q_{21} Where can I buy a hydrangea?

Q_{22} Where can I buy a cactus?

Q_{23} Where can I buy an iPad?

To alleviate the lexical chasm, we can redefine $u(Q_i, Q_r)$ to be the similarity measure proposed by (Mihalcea et al., 2006) as follows:

$$mcs(Q_i, Q_r) = \frac{\sum\limits_{w \in \{Q_i\}} maxSim(w, Q_r) * idf(w)}{\sum\limits_{w \in \{Q_i\}} idf(w)} + \frac{\sum\limits_{w \in \{Q_r\}} maxSim(w, Q_i) * idf(w)}{\sum\limits_{w \in \{Q_r\}} idf(w)}$$

Since scaling factors are immaterial for ranking, we have ignored the normalization constant contained in the original measure. For each word $w \in Q_i$, $maxSim(w, Q_r)$ computes the maximum semantic similarity between w and any word $w_r \in Q_r$. The similarity scores are weighted by the corresponding *idf*'s, and normalized. A similar score is computed for each word $w \in Q_r$. The score computed by $maxSim$ depends on the actual function used to compute the word-to-word semantic similarity. In this paper, we evaluated four of the knowledge-based measures explored in (Mihalcea et al., 2006): *wup* (Wu and Palmer, 1994), *res* (Resnik, 1995), *lin* (Lin, 1998), and *jcn* (Jiang and Conrath, 1997).

4 Supervised Learning for Question Ranking

Cosine similarity, henceforth referred as *cos*, treats questions as bags-of-words. The meta-measure proposed in (Mihalcea et al., 2006), henceforth called *mcs*, treats questions as bags-of-concepts. Both *cos* and *mcs* ignore the syntactic relations between the words in a question, and therefore may miss important structural information. In the next three sections we describe a set of structural features that we believe are relevant for judging question similarity. These and other types of features will be integrated in an SVM model for ranking, as described later in Section 4.4.

4.1 Matching the Focus Words

If we consider the question Q_{24} below as reference, question Q_{26} will be deemed more useful than Q_{25} when using *cos* or *mcs* because of the higher relative lexical and conceptual overlap with Q_{24}. However, this is contrary to the actual ordering $\langle Q_{25} \succ Q_{26} | Q_{24} \rangle$, which reflects the fact that Q_{25}, which expects the same answer type as Q_{24}, should be deemed more useful than Q_{26}, which has a different answer type.

Q_{24} What are some good thriller *movies*?

Q_{25} What are some thriller *movies* with happy ending?

Q_{26} What are some good *songs* from a thriller movie?

The analysis above shows the importance of using the answer type when computing the similarity between two questions. However, instead of relying exclusively on a predefined hierarchy of answer types, we identify the *question focus* of a question, defined as the set of maximal noun phrases in the question that corefer with the expected answer (Bunescu and Huang, 2010a). Focus nouns such as *movies* and *songs* provide more discriminative information than general answer types such as *products*. We use answer types only for questions such as Q_{27} or Q_{28} below that lack an explicit question focus. In such cases, an artificial question focus is created from the answer type (e.g. *location* for Q_{27}, or *method* for Q_{28}).

Q_{27} *Where* can I buy a good coffee maker?

Q_{28} *How* do I make a pizza?

Let f_i and f_r be the focus words corresponding to questions Q_i and Q_r. We introduce a focus feature ϕ_f, and set its value to be equal with the similarity between the focus words:

$$\phi_f(Q_i, Q_r) = wsim(f_i, f_r) \tag{1}$$

We use *wsim* to denote a generic word meaning similarity measure (e.g. *wup*, *res*, *lin* or *jcn*). When computing the focus feature, the non-focus word "movie" in Q_{26} will not be compared with the focus word "movies" in Q_{24}, and therefore Q_{26} will have a lower value for this feature than Q_{25}, i.e. $\phi_f(Q_{26}, Q_{24}) < \phi_f(Q_{25}, Q_{24})$.

4.2 Matching the Main Verbs

In addition to the question focus, the *main verb* of a question can also provide key information in estimating question-to-question similarity. We define the main verb to be the content verb that is highest in the dependency tree of the question, e.g. *buy* for Q_{27}, or *make* for Q_{28}. If the question does not contain a content verb, the main verb is defined to be the highest verb in the dependency tree, as for example *are* in Q_{24} to Q_{26}. The utility of a question's main verb in judging its similarity to other questions can be seen more clearly in the questions below, where Q_{29} is the reference:

Q_{29} How can I *transfer* music from iTunes to my iPod?

Q_{30} How can I *upload* music to my iPod?

Q_{31} How can I *play* music in iTunes?

The fact that *upload*, as the main verb of Q_{30}, is more semantically related to *transfer* is essential in deciding that $\langle Q_{30} \succ Q_{31} | Q_{29} \rangle$, i.e. Q_{30} is more useful than Q_{31} to Q_{29}.

Let v_i and v_r be the main verbs corresponding to questions Q_i and Q_r. We introduce a main verb feature ϕ_v as follows:

$$\phi_v(Q_i, Q_r) = wsim(v_i, v_r) \tag{2}$$

If Q_{29} is considered as reference question, it is expected that the main verb feature for question Q_{30} will have a higher value than the main verb feature for Q_{31}, i.e. $\phi_f(Q_{31}, Q_{29}) < \phi_f(Q_{30}, Q_{29})$.

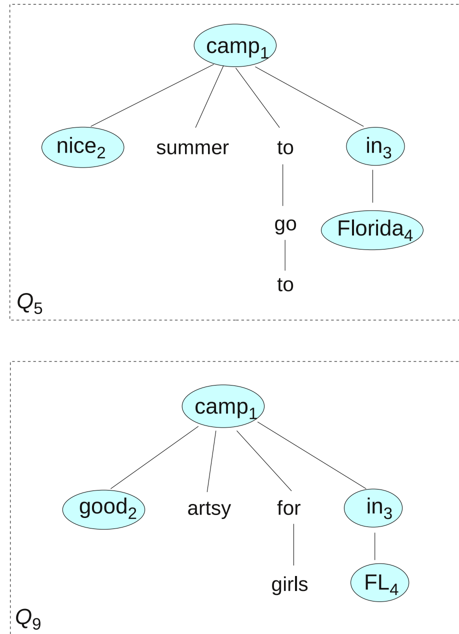

Figure 2: Matched dependency trees.

4.3 Matching the Dependency Trees

The question focus and the main verb are only two of the nodes in the syntactic dependency tree of a question. In general, all the words in a question are important when judging its semantic similarity with another question. We therefore propose a more general feature that exploits the dependency structure of the question and, in doing so, it also considers all the words in the question, like *cos* and *mcs*. For any given question we initially ignore the direction of the dependency arcs and change the question dependency tree to be rooted at the focus word, as illustrated in Figure 2 for questions Q_5 and Q_9. Interrogative patterns such as "What is" or "Are there any" are automatically eliminated from the dependency trees. We define the dependency tree similarity between two questions Q_i and Q_r to be a function of similarities $wsim(v_i, v_r)$ computed between aligned nodes $v_i \in Q_i$ and $v_r \in Q_r$. The nodes of two dependency trees are aligned through a function $\texttt{MaxMatch}(u_i.\mathcal{C},\ u_r.\mathcal{C})$ that takes two sets of children nodes as arguments, one from Q_i and one from Q_r, and finds the maximum weighted bipartite matching between $u_i.\mathcal{C}$ and $u_r.\mathcal{C}$. Given two children nodes $v_i \in u_i.\mathcal{C}$ and $v_r \in u_r.\mathcal{C}$, the weight of a potential matching between v_i and v_r is defined simply as $wsim(v_i, v_r)$. MaxMatch($u_i.\mathcal{C}$, $u_r.\mathcal{C}$) is furthermore constrained to match only nodes that have compatible part-of-speech tags (e.g. nouns are matched to nouns, verbs are matched to verbs), and children nodes that have the same head-modifier relationship with their parents (i.e. they are both heads, or they are both dependents of their parents). Table 3 shows the recursive algorithm used

TreeMatch(u_i, u_r)
[In]: Two dependency tree nodes u_i, u_r. [Out]: A set of node pairs $\mathcal{M}$.
1. **set** $\mathcal{M} \leftarrow \{(u_i, u_r)\}$ 2. **for each** $(v_i, v_r) \in$ MaxMatch($u_i.\mathcal{C}, u_r.\mathcal{C}$): 3. **set** $\mathcal{M} \leftarrow \mathcal{M} \cup$ TreeMatch(v_i, v_r) 4. **return** $\mathcal{M}$

Table 3: Dependency Tree Matching.

for finding a matching between two question dependency trees rooted at the focus words. The initial arguments of the algorithm are the two focus words $u_i = f_i$ and $u_r = f_r$. Thus, the pair (f_i, f_r) is the first pair of nodes to be added to the matching $\mathcal{M}$ in step 1. In the next step, we compute the maximum weighted matching between the children nodes $u_i.\mathcal{C}$ and $u_r.\mathcal{C}$, and recursively call the matching algorithm on pairs of matched nodes (v_i, v_r) from $\mathcal{M}$. The algorithm stops when MaxMatch returns an empty matching, which may happen when reaching leaf nodes, or when no pair of children nodes has compatible POS tags, or child-parent dependencies. Figure 2 shows the results of applying the tree matching algorithm on questions Q_5 and Q_9. Matched nodes share the same index and are shown in circles, whereas unmatched nodes are shown in italics.

We introduce a new feature $\phi_t(Q_i, Q_r)$ whose value is defined as the dependency tree similarity between questions Q_i and Q_r. Once the optimum matching $\mathcal{M}(Q_i, Q_r)$ between dependency trees has been found, $\phi_t(Q_i, Q_r)$ is computed as the normalized sum of the similarities between pairs of matched nodes v_i and v_r, as shown in Equations 3 and 4 below. When computing the similarity between two matched nodes, we factor in the similarities between corresponding pairs of words on the paths $f_i \rightsquigarrow v_i$, $f_r \rightsquigarrow v_r$ between the focus words f_i, f_r and the nodes v_i, v_r, as shown in Equation 5. This has the effect of reducing the importance of words that are farther away from the focus word in the dependency tree.

$$\phi_t(Q_i, Q_r) = \frac{sim(Q_i, Q_r)}{\sqrt{sim(Q_i, Q_i) sim(Q_r, Q_r)}} \quad (3)$$

$$sim(Q_i, Q_r) = \sum_{(v_i, v_r) \in \mathcal{M}(Q_i, Q_r)} sim(f_i \rightsquigarrow v_i, f_r \rightsquigarrow v_r) \quad (4)$$

$$sim(u_1 \rightsquigarrow u_n, v_1 \rightsquigarrow v_n) = \prod_{i=1}^{n} wsim(u_i, v_i) \quad (5)$$

If the word similarity function is normalized and defined to return 1 for identical words, the normalizer in Equation 3 becomes equivalent with $\sqrt{|Q_i||Q_r|}$. Thus, words that are left unmatched implicitly decrease the dependency tree similarity.

4.4 An SVM Model for Ranking Questions

We consider learning a *usefulness* function $u(Q_i, Q_r)$ of the following general, linear form:

$$u(Q_i, Q_r) = \mathbf{w}^T \phi(Q_i, Q_r) \quad (6)$$

The vector $\phi(Q_i, Q_r)$ is defined to contain the following generic features:

1. $\phi_f(Q_i, Q_r)$ = the semantic similarity between focus words, as described in Section 4.1.
2. $\phi_v(Q_i, Q_r)$ = the semantic similarity between main verbs, as described in Section 4.2.
3. $\phi_t(Q_i, Q_r)$ = the semantic similarity between the dependency trees, as described in Section 4.3.
4. $cos(Q_i, Q_r)$ = the cosine similarity between the two questions, as described in Section 3.
5. $mcs(Q_i, Q_r)$ = the bag-of-concepts similarity between the two questions, as described in Section 3.

Each of the generic features ϕ_f, ϕ_v, ϕ_t, and mcs corresponds to four actual features, one for each possible choice of the word similarity function $wsim$ (i.e. *wup*, *res*, *lin* or *jcn*). An additional pair of features is targeted at questions containing locations:

6. $\phi_l(Q_i, Q_r)$ = 1 if both questions contain locations, 0 otherwise.

7. $\phi_d(Q_i, Q_r)$ = the normalized geographical distance between the locations in Q_i and Q_r, 0 if $\phi_l(Q_i, Q_r)$ = 0.

Given two location names, we first find their latitude and longitude using Google Maps, and then compute the spherical distance between them using the *haversine formula*.

The corresponding parameters $\mathbf{w}$ will be trained on pairs from one of the partially ordered datasets described in Section 2. We use the kernel version of the large-margin ranking approach from (Joachims, 2002) which solves the optimization problem in Figure 3 below. The aim of this formulation is to find a

$$
\begin{aligned}
&\text{minimize:} \\
&\quad J(w, \xi) = \tfrac{1}{2}\|\mathbf{w}\|^2 + C \sum \xi_{rij} \\
&\text{subject to:} \\
&\quad \mathbf{w}^T\phi(Q_i, Q_r) - \mathbf{w}^T\phi(Q_j, Q_r) \geq 1 - \xi_{rij} \\
&\quad \xi_{rij} \geq 0 \\
&\quad \forall Q_r, Q_i, Q_j \in \mathcal{D}, \langle Q_i \succ Q_j | Q_r \rangle
\end{aligned}
$$

Figure 3: SVM ranking optimization problem.

weight vector $\mathbf{w}$ such that 1) the number of ranking constraints $u(Q_i, Q_r) \geq u(Q_j, Q_r)$ from the training data $\mathcal{D}$ that are violated is minimized, and 2) the ranking function $u(Q_i, Q_r)$ generalizes well beyond the training data. The learned $\mathbf{w}$ is a linear combination of the feature vectors $\phi(Q_i, Q_r)$, which makes it possible to use kernels.

5 Experimental Evaluation

We use the four question ranking datasets described in Section 2 to evaluate the three similarity measures cos, mcs, and ϕ_t, as well as the SVM ranking model. We report one set of results for each of the four word similarity measures *wup*, *res*, *lin* or *jcn*. Each question similarity measure is evaluated in terms of its accuracy on the set of ordered pairs, and the performance is averaged between the two annotators for the *Simple* and *Complex* datasets. If $\langle Q_i \succ Q_j | Q_r \rangle$ is a relation specified in the annotation, we consider the tuple $\langle Q_i, Q_j, Q_r \rangle$ correctly classified if and only if $u(Q_i, Q_r) > u(Q_j, Q_r)$, where u is the question similarity measure. We used the SVM^{light} [2] implementation of ranking SVMs, with a cubic kernel and the standard parameters. The SVM ranking model was trained and tested using 10-fold cross-validation, and the overall accuracy was computed by averaging over the 10 folds.

We used the NLTK [3] implementation of the four similarity measures *wup*, *res*, *lin* or *jcn*. The *idf* values for each word were computed from frequency counts over the entire Wikipedia. For each question, the *focus* is identified automatically by an SVM tagger trained on a separate corpus of 2,000 questions manually annotated with focus information (Bunescu and Huang, 2010a). The SVM tagger uses a combination of lexico-syntactic features and a quadratic kernel to achieve a 93.5% accuracy in a 10-fold cross validation evaluation on the 2,000 questions. The head-modifier dependencies were derived automatically from the syntactic parse tree using the head finding rules from (Collins, 1999). The syntactic tree is obtained using Spear [4], a syntactic parser which comes pre-trained on an additional treebank of questions. The *main verb* of a question is identified deterministically using a breadth first traversal of the dependency tree.

The overall accuracy results presented in Table 4 show that the SVM ranking model obtains by far the best performance on both datasets, a substantial 10% higher than cos, which is the best performing unsupervised method. The random baseline – assigning a random similarity value to each pair of questions – results in 50% accuracy. Even though its use of word senses was expected to lead to superior results, *mcs* does not perform better than *cos* on this dataset. Our implementation of *mcs* did however perform better than *cos* on the Microsoft paraphrase corpus (Dolan et al., 2004). One possible reason for this behavior is that *mcs* seems to be less resilient than *cos* to differences in question length. Whereas the Microsoft paraphrase corpus was specifically designed such that "the length of the shorter of the two sentences, in words, is at least 66% that of the longer" (Dolan and Brockett, 2005), the question ranking datasets place no constraints on the lengths of the

[2] svmlight.joachims.org
[3] www.nltk.org
[4] www.surdeanu.name/mihai/spear

Question Dataset	cos	*wup*		*res*		*lin*		*jcn*		*SVM*
		mcs	ϕ_t	mcs	ϕ_t	mcs	ϕ_t	mcs	ϕ_t	
Simple	73.7	69.1	69.4	71.3	71.8	70.8	69.8	71.9	71.7	**82.1**
Complex	72.6	64.1	69.6	66.0	71.5	66.9	69.1	69.4	71.0	**82.5**

Table 4: Pairwise accuracy results.

Dataset	all	$-\phi_f$	$-\phi_v$	$-\phi_t$	$-\phi_{l,d}$	$-cos$	$-mcs$	$-\phi_{f,t}$
Simple	**82.1**	79.3	82.0	80.2	81.5	80.3	81.4	78.5
Complex	**82.5**	81.3	81.3	78.7	81.8	79.2	81.8	77.4

Table 5: Ablation results.

questions. However, even though by themselves the meaning aware mcs and the structure-and-meaning aware ϕ_t do not outperform the bag-of-words cos, they do help in increasing the performance of the SVM ranking model, as can be inferred from the corresponding columns in Table 5. The table shows the results of ablation experiments in which all but one type of features are used. The results indicate that all types of features are useful, with significant contributions being brought especially by cos and the focus related features $\phi_{f,t}$.

The measures investigated in this paper are all compositional and reduce the similarity computations to word level. The following question patterns illustrate the need to design more complex similarity measures that take into account the context of every word in the question:

P_1 Where can I find a job around $\langle$*City*$\rangle$?

P_2 What are some famous people from $\langle$*City*$\rangle$?

P_3 What is the population of $\langle$*City*$\rangle$?

Below are three instantiations of the first question pattern:

Q_{32} Where can I find a job around Anaheim, CA?

Q_{33} Where can I find a job around Los Angeles?

Q_{34} Where can I find a job around Vista, CA?

If we take Q_{32} as reference question, the fact that the distance between Los Angeles and Anaheim is smaller than the distance between Vista and Anaheim leads the ranking system to rank Q_{33} as more useful than Q_{34} with respect to Q_{32}, which is the expected result. The preposition "around" from the city context in the first pattern is a good indicator that proximity relations are relevant in this case. When the same three cities are used for instantiating the other two patterns, it can be seen that the proximity relations are no longer as relevant for judging the relative usefulness of questions.

6 Future Work

We plan to integrate context dependent word similarity measures into a more robust question utility function. We also plan to make the dependency tree matching more flexible in order to account for paraphrase patterns that may differ in their syntactic structure. The questions that are posted on community QA sites often contain spelling or grammatical errors. Consequently, we will work on interfacing the question ranking system with a separate module aimed at fixing orthographic and grammatical errors.

7 Related Work

The question rephrasing subtask has spawned a diverse set of approaches. (Hermjakob et al., 2002) derive a set of phrasal patterns for question reformulation by generalizing surface patterns acquired automatically from a large corpus of web documents. The focus of the work in (Tomuro, 2003) is on deriving reformulation patterns for the interrogative part of a question. In (Jeon et al., 2005), word translation probabilities are trained on pairs of semantically similar questions that are automatically extracted from an FAQ archive, and then used in a language model that retrieves question reformulations. (Jijkoun and de Rijke, 2005) describe an FAQ question retrieval system in which weighted combinations of similarity functions corresponding to questions, existing answers, FAQ titles and pages are computed using a vector space model. (Zhao et al., 2007) exploit the Encarta logs to automatically extract clusters containing question paraphrases and further train a perceptron to recognize question paraphrases inside each cluster based on a combination of lexical, syntactic and semantic similarity features. More recently, (Bernhard and Gurevych, 2008) evaluated various string similarity measures and vector space based similarity measures on the task of retrieving question paraphrases from the WikiAnswers repository. The aim of the question search task presented in (Duan et al., 2008) is to return questions that are semantically equivalent or close to the queried question, and is therefore similar to our question ranking task. Their approach is evaluated on a dataset in which questions are categorized either as relevant or irrelevant. Our formulation of question ranking is more general, and in particular subsumes the annotation of binary question categories such as relevant vs. irrelevant, or paraphrases vs. non-paraphrases. Moreover, we are able to exploit the annotated utility relations as supervision in a learning for ranking approach, whereas (Duan et al., 2008) use the annotated dataset to tune the 3 parameters of a mostly unsupervised approach. The question ranking task was first formulated in (Bunescu and Huang, 2010b), where an initial version of the dataset was also described. In this paper, we introduce 4 versions of the dataset, a more general meaning and structure aware similarity measure, and a supervised model for ranking that substantially outperforms the previously proposed utility measures.

8 Conclusion

We presented a supervised learning approach to the question ranking task in which previously known questions are ordered based on their relative utility with respect to a new, reference question. We created four versions of a dataset of 60 groups of questions [5], each annotated with a partial order relation reflecting the relative utility of questions inside each group. An SVM ranking model was trained on the dataset and evaluated together with a set of simpler, unsupervised question-to-question similarity models. Experimental results demonstrate the importance of using structure and meaning aware features when computing the relative usefulness of questions.

[5] The dataset will be made publicly available.

Acknowledgments

We would like to thank the anonymous reviewers for their insightful comments.

References

Ricardo Baeza-Yates and Berthier Ribeiro-Neto. 1999. *Modern Information Retrieval*. ACM Press, New York.

Delphine Bernhard and Iryna Gurevych. 2008. Answering learners' questions by retrieving question paraphrases from social Q&A sites. In *EANL '08: Proceedings of the Third Workshop on Innovative Use of NLP for Building Educational Applications*, pages 44–52, Morristown, NJ, USA. Association for Computational Linguistics.

Razvan Bunescu and Yunfeng Huang. 2010a. Towards a general model of answer typing: Question focus identification. In *Proceedings of The 11th International Conference on Intelligent Text Processing and Computational Linguistics (CICLing 2010), RCS Volume*, pages 231–242.

Razvan Bunescu and Yunfeng Huang. 2010b. A utility-driven approach to question ranking in social QA. In *Proceedings of The 23rd International Conference on Computational Linguistics (COLING 2010)*, pages 125–133.

Michael Collins. 1999. *Head-driven Statistical Models for Natural Language Parsing*. Ph.D. thesis, University of Pennsylvania.

William B. Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*, pages 9–16.

Bill Dolan, Chris Quirk, and Chris Brockett. 2004. Unsupervised construction of large paraphrase corpora: Exploiting assively parallel news sources. In *Proceedings of The 20th International Conference on Computational Linguistics (COLING'04)*, page 350.

Huizhong Duan, Yunbo Cao, Chin-Yew Lin, and Yong Yu. 2008. Searching questions by identifying question topic and question focus. In *Proceedings of ACL-08: HLT*, pages 156–164, Columbus, Ohio, June.

Ulf Hermjakob, Abdessamad Echihabi, and Daniel Marcu. 2002. Natural language based reformulation resource and web exploitation for question answering. In *Proceedings of TREC-2002*.

Jiwoon Jeon, W. Bruce Croft, and Joon Ho Lee. 2005. Finding similar questions in large question and answer archives. In *Proceedings of the 14th ACM international conference on Information and knowledge management (CIKM'05)*, pages 84–90, New York, NY, USA. ACM.

J.J. Jiang and D.W. Conrath. 1997. Semantic similarity based on corpus statistics and lexical taxonomy. In *Proceedings of the International Conference on Research in Computational Linguistics*, pages 19–33.

Valentin Jijkoun and Maarten de Rijke. 2005. Retrieving answers from frequently asked questions pages on the Web. In *Proceedings of the 14th ACM international conference on Information and knowledge management (CIKM'05)*, pages 76–83, New York, NY, USA. ACM.

Thorsten Joachims. 2002. Optimizing search engines using clickthrough data. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2002)*, Edmonton, Canada.

Dekang Lin. 1998. An information-theoretic definition of similarity. In *Proceedings of the Fifteenth International Conference on Machine Learning (ICML '98)*, pages 296–304, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Rada Mihalcea, Courtney Corley, and Carlo Strapparava. 2006. Corpus-based and knowledge-based measures of text semantic similarity. In *Proceedings of the 21st national conference on Artificial intelligence (AAAI'06)*, pages 775–780. AAAI Press.

Dan I. Moldovan, Marius Pasca, Sanda M. Harabagiu, and Mihai Surdeanu. 2002. Performance issues and error analysis in an open-domain question answering system. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 33–40, Philadelphia, PA, July.

John M. Prager. 2006. Open-domain question-answering. *Foundations and Trends in Information Retrieval*, 1(2):91–231.

Philip Resnik. 1995. Using information content to evaluate semantic similarity in a taxonomy. In *IJCAI'95: Proceedings of the 14th international joint conference on Artificial intelligence*, pages 448–453, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Noriko Tomuro. 2003. Interrogative reformulation patterns and acquisition of question paraphrases. In *Proceedings of the Second International Workshop on Paraphrasing*, pages 33–40, Morristown, NJ, USA. Association for Computational Linguistics.

Zhibiao Wu and Martha Palmer. 1994. Verbs semantics and lexical selection. In *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, pages 133–138, Morristown, NJ, USA. Association for Computational Linguistics.

Shiqi Zhao, Ming Zhou, and Ting Liu. 2007. Learning question paraphrases for QA from Encarta logs. In *Proceedings of the 20th international joint conference on Artifical intelligence (IJCAI'07)*, pages 1795–1800, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.