Category-specific Models for Ranking Effective Paraphrases in Community Question Answering

Alejandro Figueroa[a,b], Günter Neumann[c,*]

[a] *Yahoo! Research Latin America, Blanco Encalada 2120, Santiago, Chile*
[b] *Escuela de Ingeniería Informática, Universidad Diego Portales, Santiago, Chile*
[c] *DFKI GmbH, Stuhlsatzenhausweg 3, Campus D3_2, D-66123 Saarbrücken, Germany*

Abstract

Platforms for community-based Question Answering (cQA) are playing an increasing role in the synergy of information-seeking and social networks. Being able to categorize user questions is very important, since these categories are good predictors for the underlying question goal, viz. informational or subjective. Furthermore, an effective cQA platform should be capable of detecting similar past questions and relevant answers, because it is known that a high number of best answers are reusable. Therefore, question paraphrasing is not only a useful but also an essential ingredient for effective search in cQA. However, the generated paraphrases do not necessarily lead to the same answer set, and might differ in their expected quality of retrieval, for example, in their power of identifying and ranking best answers higher.

We propose a novel category-specific learning to rank approach for effectively ranking paraphrases for cQA. We describe a number of different large-scale experiments using logs from Yahoo! Search and Yahoo! Answers, and demonstrate that the subjective and objective nature of cQA questions dramatically affect the recall and ranking of past answers, when fine-grained category information is put into its place. Then, category-specific models are able to adapt well to the different degree of objectivity and subjectivity of each category, and the more specific the models are, the better the results, especially when benefiting from effective semantic and syntactic features.

Keywords: community-based Question Answering; learning to rank; question paraphrases; question categories

1. Introduction

Web browsing has become a de facto standard for information seeking in our daily life. Search engines play a key role here in bridging the gap between the information seekers and the massive collection of web data. Understanding web queries for guiding the search effectively is a difficult task, since distinct users do not only formulate their queries with different terminologies, intents, and linguistic patterns, but they also exhibit assorted browsing behaviors. This

*Corresponding author; phone: +49 (0)681 / 857 75-0
Email addresses: afiguero@yahoo-inc.com, alejandro.figueroa@mail.udp.cl (Alejandro Figueroa), neumann@dfki.de (Günter Neumann)

challenging nature together with the goal of enhancing and personalizing search experience encourage developers of web search engines to investigate more intelligent algorithms for understanding and satisfying the requests of their users.

The advances made by search engines, i.e., offering more powerful services, have given their users the chance of reaching more specific and ambitious goals, and actually, have caused them to become more audacious when prompting queries. With the advent of social media, users are now more and more likely to enter complex and complete questions instead of few keywords, especially when they are targeting at precise information needs. Nonetheless, answers to these complex questions are hardly found in short text fragments within web pages or across full documents, because they require the analysis, understanding, and synthesis of several documents and world knowledge. For example, complex questions aim at current events (e.g. "*Who will win this Australian Open?*"), finding sentiments of the general public about something or someone (e.g. "*What is the coziest Starbucks in Manhattan?*"), at subjective opinions regarding particular topics (e.g., contrasting different products), which, at the moment of searching, do not necessarily exist on the web in the form of conventional web documents (e.g., "*How do you envision tablets in the year 2020?*")

Since these kinds of information needs are difficult to fulfill by means of traditional information retrieval techniques, web users take advantage of community Question Answering (cQA) services for getting help from other individuals, who know or can readily produce satisfactory precise answers, or like in many cases, can provide help by conducting opinion polls and surveys. In a nutshell, these platforms (e.g., Yahoo! Answers) are the synergy of a information-seeking and a social network [1], where members can post any kind of question, either simple, complex or detailed, or questions about opinions. In a similar way, posted questions can receive several responses from multiple members, which can not only be supplementary or complementary to each other, but also reflect different sentiments and aspects. When taking part in this network, members additionally provide social capital: rate the answers' quality (via positive/negative votes, thumbs-up/thumbs-down, etc.) and post comments. In summary, the information-seeking perspective of a cQA provides arbitrary members with content, motivating them to take part in asking and responding questions, especially when the experience of social interactions is positive; while the social network perspective causes members to engage in social activities [1].

Through these social interactions, members share their knowledge so as to construct a valuable, rapidly growing and massive archive of questions and answers. Notably, one attractive part of these repositories yields a large quantity of diverse word-of-mouth tips (e.g., "*How to get rid of eye strain?*" and "*Teach my cat to use the toilet*"), insights and solutions to many common questions and daily problems that people may face (e.g., "*Removing cooked on grease from pans?*"). CQA services are usually organized in categories, which are selected by members when submitting new questions. These categories are later utilized for locating contents on topics of interest. In a category to which only social activity is attached, fewer members respond to questions, resulting in a small average number of answers per question causing a low rate of user satisfaction. This is in contrast to a category where social activities and information-seeking activities co-occur: the amount of answers is average or above [1].

Recent studies have unveiled that this synergy is also projected into the relationship between categories and question intents [2]. More precisely, they revealed that categories are good predictors of question goals. Although the number of types of intents varies from one approach to another [2, 3, 4], most studies agree on two main types of ends [2]: informational (i.e., objective or information-seeking) and subjective (i.e., social, opinions or conversational). The following Yahoo! Answers categories exemplify this contrast: "*Polls & Surveys*" and "*Religion & Spirituality*" embrace almost solely subjective questions, while this kind of intent covers 70% of "*Singles & Dating*", 27.27% of "*Health*" and 16.17% of "*Science & Mathematics*", only.

Due to several reasons (e.g., system saturation [5] or bad question formulations [6]), it has been observed that about 15% of all incoming questions in English go unresolved, poorly answered or never satisfactorily resolved in Yahoo! Answers [7]. Thus, an effective cQA platform should be capable of detecting similar past questions and relevant answers. Practical solutions would involve asking members for rephrasing a question [6], suggesting alternative questions [8], or offering past answers, since at least 78% of best answers are reusable [4, 9]. However, the lexical gap between past and new questions is the main obstacle to reuse these best answers (e.g., "*Remove pimples?*" and "*How to get rid of acne*"), thus some strategies have tried to combine social and textual (e.g., semantic and syntactic) features as a means of tackling lexical mismatches, cf. [9, 10, 11, 12].

A promising approach to improve the effectiveness of search in cQA by means of automatic identification of question paraphrases has been proposed by [13]. The core idea is to use the user generated questions of a cQA along with search engine query logs to automatically formulate effective questions or paraphrases in order to improve search in cQA. [14] have further elaborated this idea into the direction of generation of new questions from queries. A major advantage of such a query-to-question expansion approach for cQA is that it can help to retrieve more related results from cQA archives and hence, can improve the recall.

The automatic generation of paraphrases is a useful means to improve the search for finding best answers in cQA. But the generated paraphrases (although they might "mean" the same) do not necessarily lead to the same answer set, and hence, it might be that they differ in the expected retrieval quality of identifying and ranking best answers high. Thus, it makes sense to rank the generated paraphrases, so as to provide evidence according to recall and the position of the best answer of a paraphrase, i.e., its mean reciprocal rank (MRR). This is the major motivation behind our approach of computing *effective paraphrases*. An effective paraphrase is a reformulation of the posted question that narrows the lexical gap the best, i.e., an alternative formulation of a user question that can retrieve more past answers to the new question, or can rank past answers higher within the fetched set (see examples in table 1). In [15], we presented a first learning to rank approach based on general-purpose models that is able to determine effective question paraphrases by exploiting search engine query logs and connections to cQA, however, without taking into account question category-specific information. This work extends our earlier work on several innovative aspects:

1. We empirically demonstrate that the subjective and objective nature of cQA questions dramatically affect the recall and ranking of past answers. Since categories and question intents are closely related, we construct

category-specific learning to rank models (i.e., SVMRank) for paraphrase ranking, showing that the retrieval and ranking from social media can be improved when category information is put in place.

2. Since we carry out experiments on a large data-set of automatically annotated question paraphrases harvested from Yahoo! Answers and Yahoo! Search logs, we are able to conduct experiments not only on broad, but also on fine-grained question categories. Specifically, we consider the three levels of granularity supplied by the Yahoo! Answers question taxonomy.
3. In addition, we study the impact on our category-specific models of Natural Language Processing (NLP) information in two ways: a) we show that enriching question categorization with Wh-question typification enhances the performance; and b) our models are built largely on the basis of effective semantic and syntactic properties.

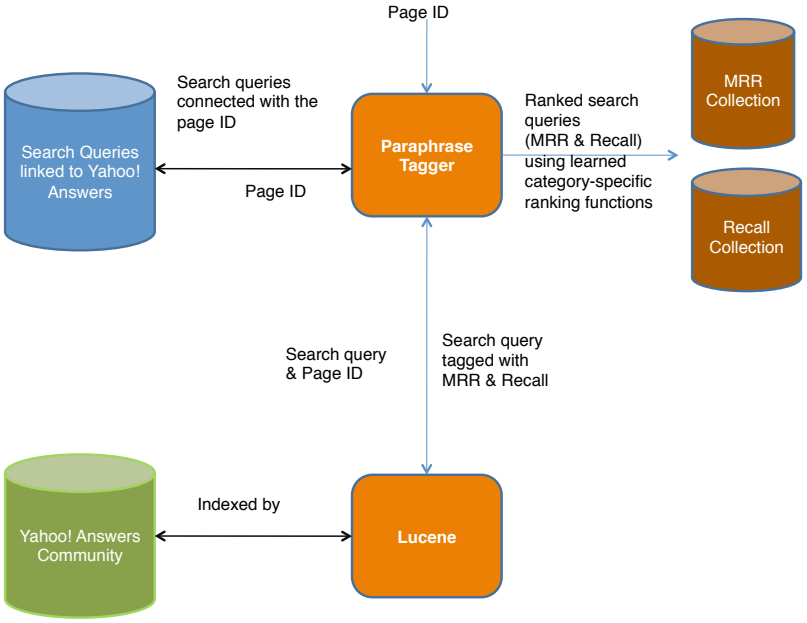

Figure 1: Major components and control flow for both, the training and application phase.

The core idea of our method is as follows (see Fig. 1). Given a huge collection of query logs from Yahoo! Search, we extract all pairs consisting of a query and a title, where at least one user click links the query with a title from Yahoo! Answers. Note that the title is the user entered question of the answer web page together with the category selected by the user.[1] We further cluster these pairs into groups, where each group consists of all query-title pairs with same title and category. We interpret each group (including the title) as a set of paraphrases of the same underlying question and category. Note that each title is associated with an answer web page, and so also its paraphrases. This way we obtain a huge collection of 32 million answer web pages and their associated question paraphrases completely

[1] Actually, it is mandatory, that if a user enters a question to Yahoo! Answers, he or she also selects a category from a given set.

automatically. We construct a full text search index from the answers of this corpus using Lucene such that we consider each individual answer as a document.

In a next step we assign to each paraphrase a recall and MRR value which are automatically computed by querying each paraphrase (including the original user question) to the indexed collection. We achieve this, by automatically assessing a paraphrase by sending it to Lucene and checking its recall and the MRR of the highest ranked best answer (we keep the answer page ID, which allows as to assign all answers retrieved by Lucene to its corresponding answer page in the aligned corpus). The recall is computed by accounting for the number of answers fetched from the related Yahoo! Answers page.

In this way, we are able to automatically sort all paraphrases sets of our corpus according to recall and MRR independently, cf. table 1 for an example. Furthermore, we can extract features and learn two separate learning to rank (SVMRank) models for each category, one for ranking new paraphrases according to recall and one for ranking them in congruence with MRR. The example in table 1 actually illustrates that the original user question "*How does direct deposit of tax refund work for joint filers ?*" receives a lower MRR than the automatically determined paraphrase "*must you have a joint account for a direct deposit*".

The category information is only used for learning the ranking models. The paraphrases of a new query are assumed to be ranked before Lucene is called by the category-specific models in the application phase, i.e., the category is a parameter for the paraphrase identification process, but not for the retrieval process. We are using recall and MRR for measuring the quality of a paraphrase, and they have been computed completely automatically relative to Yahoo! Answers' answers. Hence, a paraphrase is better than another one if it has a higher recall or MRR potential for retrieving pages with best answers. We compute paraphrases and do the ranking because we do want to improve search in cQA in the sense that we "manipulate" a user query in order to find a better paraphrase, and better means, better with respect to recall and MRR.

The results of our experiments show that fine-grained category-specific models can assist in boosting the retrieval and ranking from cQA archives. Without the generation and ranking of paraphrases of a user query, sending the query to Lucene would just realize a simple IR scenario: send a query and receive documents. However, with the help of our automatically learned ranking models, we generate paraphrases (realized by means of available links between Yahoo! search queries and Yahoo! answers) and rank them by using the available category-specific ranking functions. Since the retrieval results obtained by using these category-specific queries are much better compared to the retrieval results found with the original user queries, the ranking function has helped to identify category-specific lexical information very reliable. In particular, our outcomes indicate that these specific models capture the different degrees of objectivity and subjectivity behind the distinct categories via an ad-hoc exploitation of a state-of-the-art machine learning technique in conjunction with lexical, semantic and syntactic properties. In addition, our experiments underline the positive contribution of shallow syntactic cues, i.e., Wh-question types, to this task. Thus, in other words, we can show that when the goal is of retrieving cQA answers, then generating paraphrases and ranking them using category-specific information is extremely helpful. If we have a question and if we know what it is about

(in form of a category) then the ranking of generated paraphrases leads to much better results than if we did not do it.

We don't think that this is a trivial observation, especially, when considering the huge amount of data. From a theoretical point of view, our results show that effective paraphrases are characterized by structures that can be inferred from ordered samples, and these learned structures are usable for recognizing new, unseen effective paraphrases. These structures do not only depend on the task at hand — in our case, the improvement of the recall and ranking of answers — but also on the category of the paraphrases and on the granularity of the respective categorization system. Through a wide number of large-scale experiments carried out on real-life large-scale data collections, we show, that specialized structures targeted at fine-grained categories achieve better performance than general structures that disregard categorization. Thus, knowing the question category for controlling the answer selection process is at least as important for intelligent community QA as it is for knowing the expected answer type in standard text-based QA systems.

Note that we conceive paraphrases in a broad sense, that is we do not explicitly only consider well-formulated questions (e.g., "does lack of iron cause headaches?"), but also implicit requests ("headache iron"), grammatically incorrect queries ("and headach low iron") and other semantic alternatives ("migraine headaches low iron" or "can low hemoglobin cause headaches?"). Note further, that all information stored in a web answer page is retained, which in general not only contains relevant answers, but any comment made by the community for that selected question. We think, that these issues, viz. linguistic variability, data sparseness and the impact of question categorization mentioned above are very important to define realistic test cases and to achieve robustness on real cQA data (cf. also sec. 5).

The focus of this paper is on the corpus creation process and the category-based learning to rank models. Therefore, the structure of the paper is organized as follows: section 2 presents and discusses the most relevant related work, section 3 details the corpus creation process and our category-based learning to rank models, and section 4 deals at length with our experiments. Finally, section 5 draws the main conclusions.

2. Related Work

2.1. Question Processing in cQA

Since cQA platforms have to cope with questions aimed at personal opinions and experiences, [16] proposed a cost-efficient solution built on top of an SVM trained with trigrams features which checks whether questions are objective or subjective. Similarly, the work of [3] used co-training for building an SVM approach based on text and meta-data attributes in order to group questions into three categories, viz. subjective (personal opinion), objective (factual knowledge) and social (social interaction). They found that Wh-questions (i.e., who, when, where, what and why) are more likely to bear an objective intention, whereby questions containing polite words and conversational phrases are more probable to state a subjective or social intent. Furthermore, social questions are often accumulated in some specific topics and more often be prompted by experienced members.

[2] presented a method which used a Bayesian network for learning to classify questions as informational or conversational. Their study revealed that top-level question categories are good predictors of the nature of the question. They noticed that terms such as "*how*", "*where*", "*can*" and "*I*" are often indicating informational questions, whereas terms including "*why*" and "*you*" together with phrases such as "*do you*" are more likely to signal conversational questions. Another finding regards the fact that members, who are enthusiastic about prompting conversational questions, interact more with other members than those who submit informational questions. Their study additionally showed that askers of conversational questions, contrary to askers of informational questions, are more expected to yield many responses.

CQA platforms are massive repositories of questions and answers pairs, hence past questions and/or best answers can be presented as alternative questions and/or tentative responses when askers are waiting for other members to reply their new questions. This fact motivated [8] to devise the Minimum Description Length (MDL) based tree cut model for question suggestion on restricted domains. Another strategy is due to [10, 11], which distinguished similar questions by profiting from a quadripartite network constructed with concepts distilled from the best answer picked by asker, asker profile and answerer profile with respect to a question. Although the network representation helps to identify lexical mismatches, its computational time is very demanding, especially when taking into account the dynamic nature of large-scale collections such as cQA services. Along the same line, [12] identified similar questions relevant to new queries via a reformulation of the tree kernel retrieval framework. By exploiting semantic and syntactic attributes in conjunction with answers, they narrowed the gap caused by lexical mismatches. In [17] a similar approach is presented, which recognizes the similarity of questions by computing the textual entailment between new and known questions.

The research of [6] revealed some patterns observed by unresolved questions. Their analysis showed that some categories are more probable than others to contain unanswered questions. They also showed that questions containing more subjective words are more likely to be resolved completely. Inversely, the larger the amount of polite words, the higher the likelihood that it will remain unresolved. They postulated that these questions are probable to consider troublesome experiences. Still yet, based on these findings, they found it difficult to train a high performance binary classifier.

2.2. Answer Processing in cQA

[9] presented a framework for ranking answers, where right answers to factoid questions are fetched by fusing relevance, member interaction, and community feedback information. Their framework considered various collaborating features, including number of terms, overlapping words between queries, questions and answers, the lifetime of questions and responses, askers and answerers social statistics. Their investigation revealed that it is more pertinent to top-ranked answers to be picked as "best" by the asker than to have appropriate textual characteristics. In a similar way, the retrieval technique of [18] mixed a translation-based language model for the question part with a query likelihood approach for the answer part. Our own prior work described in [15] focused on learning to rank

models to recognize effective search queries for fetching and ranking answers from cQA repositories by exploiting SVMRank [19]. We revealed that Wh-question-type based models slightly outperform general-purpose models when identifying effective paraphrases, whereas the present paper shows that category-specific models are a much better option to model the objective and subjective nature of cQA content.

In order to measure the quality of user-generated answers, two distinct sources of predictors of high-quality answers have been examined: social and textual characteristics [20]. In terms of social attributes, the most salient ones are the answerer authority and the answer rating of the asker [21], whereas for textual features the most prominent attributes are the amount of unique words, the length of the response and the number of misspellings. [22] combined both sorts of predictors, showing that both are instrumental in automatically selecting high-quality user-generated answers. In the same vein, [23] determined the answer quality on the basis of two properties: answer features and member expertise. They found out that accounting for member expertise enhances the performance. [20] pointed out that most discriminative attributes cover dimensions such as comprehensiveness, truthfulness, and practicality.

[24] investigated predictors of answer quality through a comparative study of responses across several cQA services. They discovered that the topic has a major impact on the amount of posted responses, but a modest effect on their quality. For example, entertainment questions obtain many low-quality responses in relation to other topics. Furthermore, question types influence answer quality, in particular advice (*how-to*) questions reap the best quality, while factual ones the poorest. Conversely, types have no impact on the number of answers. In general, advice questions appear to catch the most and best attention of cQA members, causing the emergence of new methods targeting exclusively at this particular type of question [25, 26, 27, 28].

The idea behind [7] is reusing resolved questions for estimating the probability of new questions to be answered by past best answers. Their strategy capitalized on Latent Dirichlet Allocation (LDA) for inferring latent topics for each category, and they compared the distribution of topics for the new and previous questions as well as the answers. Incidentally, [4] proposed taxonomies for both questions and answers. Fundamentally, their question taxonomy extended [29] by adding a social category, which comprises queries that seek interactions with people. They discovered a high correlation between answer and question types. More specifically, constant questions are more likely to target factual unique answers, while opinions get subjective answers.

3. Our Model

In this section we describe how we automatically determine our annotated data collection used for identifying question paraphrases and how it is used for learning to rank these paraphrases. This work extends our earlier study on effective paraphrase identification for cQA platforms by generating category-specific ranking models inferred from automatically acquired and annotated data, cf. [15]. The basic idea behind detecting useful paraphrases is to distinguish reformulations of a posted question that are powerful for discovering good responses across past answers in cQA repositories.

Our method starts from a question submitted to Yahoo! Answers together with its respective category assigned by the asker (cf. also sec. 1). Taking these two parameters into account, our strategy receives as input a set of paraphrases derived from the posted question. In practice, this set can be obtained via a paraphrase generation component. However, in the scope of this work, we acquire this set of paraphrases via mining Yahoo! Search query logs. Next, these paraphrases are automatically tagged in congruence with their effectiveness in ranking and fetching answers. Our approach extracts several lexical, syntactic and semantic features from the posted question and its corresponding paraphrases, which serve as building blocks of our category-specific models afterwards.

These models are grounded on a learning to rank technique named SVMRank [19]. Ranking SVM is a supervised learning approach based on Support Vector Machines. It is targeted at solving some ranking problems, this means that the training material consists of arrays of ordered items, like sets of paraphrases sorted by their effectiveness. This order is frequently denoted by an ordinal score such as a recall or MRR value. More precisely, SVMRank is a pair-wise method, meaning that it learns the order between pairs of elements in a given array of ordered items, and it aims to minimize the average amount of inversions in ranking. The learned model is then utilized for putting in order unseen lists, e.g., a new set of paraphrases for a new question.

This study focuses on exploring the efficiency of several category-specific learners in retrieving and scoring past answers for new questions. These models are specified by the question taxonomy available to the user when categorizing questions. Thus this section details the components of our approach: corpus acquisition (sec. 3.1), corpus cleaning (sec. 3.2), automatic corpus annotation (sec. 3.3) and finally, describes the features exploited by our learners (sec. 3.4).

3.1. Question-paraphrase Collection

Although statistical models for generating paraphrases exist (e.g., [30]), we preferred to extract them from search query logs as a means of broadening the sampling of potential candidates. The idea of our corpus acquisition technique is to interpret question-like search queries as potential question paraphrases. This motivation is based on the observation that if some queries result in similar click patterns, then the meanings of these queries should be similar, cf. [31]. The identification of paraphrases from search engine query logs, as we do, allows us to explore a wide range of verbalizations of paraphrases, basically, from a set of few keywords (e.g., "*headache iron*") to a complete natural language question ("*does lack of iron cause headaches?*"), and we will show that our approach can cope with this kind of linguistic variability. We consider this an important aspect to define realistic test cases and to achieve robustness on real cQA data (see also sec. 1).

Along this line, [32] pioneered the extraction of high qualitative paraphrases from general-purpose search engine query logs and utilized them for producing paraphrase patterns. They found that when several queries hit the same title, these queries are likely to be paraphrases of each other. Similarly, when a query hits several titles, paraphrases can also be found among these titles. They extracted and validated three sorts of general paraphrases from search logs and mixed them into one model: query-title, query-query and title-title paraphrases. Our work sharply differs

from [32] in that we build category-specific models and evaluate the effectiveness of question-like paraphrases in terms of ranking and recall, whereas [32] focuses their attention on validating general paraphrases. Like [32], we explicitly extract query-title pairs, where each title is the question of a corresponding Yahoo! Answers page, and as such, our approach is specifically tailored to cQA services. More precisely, we perceive a question title and its linked search engine queries as a set of paraphrases of the same underlying question. We conceive the title question as the source paraphrase, while the associated search engine queries as its alternative verbalizations. Note that each source paraphrase is entered by a cQA member when setting the discussion topic of the answer page, and this is the title that search users read when clicking the respective search result.

We firstly compile a collection of queries submitted to the Yahoo! search engine during the period of January 2011 to March 2012.[2] Since we are only interested in user queries that can be utilized to find answers in Yahoo! Answers, we only retain those elements which have at least one user click that connects the search query with any question in this cQA service. We made allowances only for questions posted to this community from June 2006 to December 2011. The difference in the time period makes sense, because some time is needed to accumulate clicks to the corresponding Yahoo! Answers pages. Overall, this step collects 155 million search engine queries corresponding to about 26 million Yahoo! Answers pages.

3.2. Corpus Cleaning and Indexing

Since we noticed that many answers posted by the members are expressed in languages different from English, we checked every answer and title contained in our collection of 26 million pages. It might be the case that the search query is expressed in English, but the related (clicked) Yahoo! Answers web page is, to a large extent, in another language. For this purpose, we use a language detector[3] to filter out non-English text.

Furthermore, given the fact that some questions were duplicated in the community, we merged these instances by means of title string matching. We also removed all pages connected with more than fifty and less than five paraphrases. Pages linked with a high number of paraphrases are not reliable and make the next step too computational demanding, while pages connected with few queries are unlikely to provide good and bad reformulations. Note that due to merging, some questions might now have multiple best answers. Here we additionally discarded pages (and their related search queries) that have no best answer.

Altogether, this yields a final corpus of about 32 million answers embodied in 6 million pages corresponding to 81 million search engine queries. We indexed this pool of 32 million answers with Lucene[4]. During the indexing process we removed all stop words by means of a list of traditional stop terms extended with some tokens that we identified as community stop words (e.g., "*yummy*", "*coz*", "*lol*", and "*y!*"). All terms were lowercased.

[2] We only consider English queries, but the whole approach only uses few language specific resources, so that the adaptation of our approach to queries from other language should not be too difficult.

[3] http://code.google.com/p/language-detection/.

[4] http://lucene.apache.org/

Recall	Posted Question	Paraphrase	MRR	Posted Question	Paraphrase
0.000	F	washing face without opening pores	0.000	F	joint tax refund on debitcard
0.071	F	best water washes for face w/ pores	0.001	F	when does direct diposit og in for taxes?
0.071	F	how to open up the pores and wash your face	0.002	F	does your name have to be on checking account
0.142	F	hot water on face to open pores			to recieve direct deposits
0.142	F	should we use luckwarm water on face	0.050	F	direct deposit tax refund non joint account
		in hot summer	0.053	T	How does direct deposit of tax refund work for
0.214	F	does cold or hot water on your			joint filers?
		face open your pores	0.111	F	direct deposit for tax returs for joint filers
0.214	F	does washing face by cold water open pores	0.333	F	will the irs direct deposit a joint return into
0.285	T	Does hot or cold water open up your pores?			a single account
		which is best to wash your face with?	0.500	F	direct deposit + joint filers
0.357	F	it cold water or hot water that opens	1.000	F	does direct deposit account have to be joint
		pores on your face?	1.000	F	must you have a joint account for a direct deposit

Table 1: Two illustrative rankings. The left table is distilled from the recall collection and it shows a ranking consisting of 9 paraphrases and 6 distinct ranking scores. The right part is taken from the MRR collection. The title of the corresponding Yahoo! Answers page is marked as T, others as F.

What is the cheapest method to get to buffalo from new york?	**How to cook Rabbit ?**
i'd look on the jetblue website. they usually have cheap plane tickets, but the tickets do get more expensive the closer you get to the trip. amtrak is also usually pretty cheap, and has a bunch of discounts (AAA, student advantage etc) that can make the trip more affordable. if you're doing a round trip from buffalo to ny to buffalo, you might consider driving to rochester...that's where i fly out of/into and a lot of times the tickets are a little cheaper. it's also a direct flight on jetblue from jfk to rochester and is only about 1 hour long. if you fly into jfk it's really easy to get public transportation into nyc. even if amtrak or greyhound are cheaper than a plane is, it might not be worth it because it would basically suck up an entire day with traveling.	You can fry it, just as you would a chicken. Here is a recipe. 1 cut up rabbit 1 egg 1 cup milk flour salt pepper oil for frying Directions: Combine egg in milk. Mix flour for dredging with salt and pepper. Heat up about one inch of oil in an electric frying pan. Dip the rabbit pieces first in the egg and milk mixture, then in the flour mixture. Fry as you would chicken until golden brown. Drain on paper towels

Table 2: Two examples of best answers. The left shows a best answer chosen by the asker; the right side a best answer selected by voting.

3.3. Corpus Annotation: Recall and MRR collections

The next step is to automatically assess each paraphrase by sending it as a query to Lucene; this way we compute its recall and the MRR (Mean Reciprocal Rank) of its highest ranked best answer. The recall is computed by accounting for the number of answers fetched from the related Yahoo! Answers page, or in the event of merged pages, from the combination of all related pages. In essence, we deemed as relevant to a paraphrase all answers posted by members to the corresponding question (page title). In this sense, relevant answers were determined by humans involved in the answering process of the target question. In the case of the MRR value, the best answer is picked by the asker or in conformity to the votes casted by community members (see table 2). In all these computations, we only considered the top 1,000 hits returned by Lucene. As a result, each paraphrase is now automatically annotated with both metrics, and we construct the recall and MRR collections as follows:

- The **recall collection** comprises all pages for which we find more than three distinct values for recall across the related paraphrases. Since this rule produced few rankings, we aggregated this set with small rankings (six paraphrases) containing three different ranking values. Eventually, this brought about an increase from 36,803 to 51,148 rankings. The final amount of paraphrases is 814,816.

- The **MRR collection** encompasses all pages for which we find more than six distinct values for MRR across the related paraphrases. This rule selected 54,848 rankings containing 1,195,974 paraphrases.

Table 1 illustrates one ranking from each collection. Since an answer page can now be perceived as a ranking of paraphrases (i.e., the search engine queries together with the title of the respective page), we can now label each ranking with the category associated by the user when posting the question to the community. In detail, the question taxonomy system used by Yahoo! Answers consists of three levels and a total of 1,660 categories, where the first-level comprises 26 distinct classes (see tables 4 and 5). Thus each ranking is connected with a third-level leaf-node of this taxonomy. For the examples in table 1, the MRR ranking was associated with the third-level category "*Business & Finance→Taxes→United States*", while the recall ranking with "*Beauty & Style→Skin & Body→Other*".

For the reminder of this paper, answers are no longer utilized, and both collections are used separately during feature extraction, training and testing.

3.4. Features

During our experiments, we took into account the following array of lexical, syntactic and semantic attributes distilled from paraphrases:

- **Bag of Words** (BoW) adds a property to the feature vector representing each term and its frequency within the paraphrase, only considering terms with a global frequency higher than an empirical threshold (see sec. 4). Similarly, bigram and trigram features are computed.

- **Part-of-speech (POS) tagging** generates features in agreement with their POS categories.[5] This attribute adds to the feature vector "*number-of*" attributes: tokens in the paraphrase, tokens tagged as NN, JJ, VB, etc. The "*number-of*" frequency counts are associated with each paraphrase.

- We capitalized on **semantic relations** provided by WordNet such as HYPERNYMS (e.g., "*hardware → store*"), HYPONYMS ("*credit → payment*"), MERONYMS ("*navy → fleet*"), ATTRIBUTES ("*high → level*"), and REGIONS ("*Toronto → Canada*"). Similarly to the "*number-of*" attributes, an element representing the frequency count of the respective type of a relation at the paraphrase level is added to the feature vector.

- Analogously, we considered collocations provided by the Oxford Dictionary in order to model some **syntactic relations** between a pair of words: FOLLOWING (e.g., "*meat → rot*") and PRECEDING VERBS ("*consume → meat*"),

[5] Using http://web.media.mit.edu/~hugo/montylingua/

QUANTIFIERS ("*slab* → *meat*"), ADVERBS ("*steadily* → *increase*"), ADJECTIVES ("*souvenir* → *mug*"), VERBS ("*fill* → *mug*"), PREPOSITION ("*increase* → *by*"), and RELATED NOUN ("*meat* → *products*").

- We used **eight string distances**[6]: JARO, JACCARD, JARO-WINKLER, FELLEGI-SUNTER, LEVENSTEIN, SMITH-WATERMAN, MONGE-ELKAN and SCALED-LEVENSTEIN. For each metric, an additional attribute represents the maximum value between two different tokens in the paraphrase.

- **Word Lemma** is a boolean property indicating whether or not both, a word and its lemma are contained in the paraphrase, e.g. "*song*" and "*songs*". We used Montylingua for the morphological analysis.

4. Experiments

All our ranking models are built on top of SVMRank, which implements a fast pairwise state-of-the-art learning to rank approach capable of dealing with large-scale data-sets [19]. In order to maintain consistency across our experiments, five-fold cross validation was conducted using the same five data random splits. It is worth highlighting that our evaluations were carried out on both collections independently: All experiments assessing MRR are conducted on the MRR collection, while all experiments evaluating recall are carried out on the recall collection.[7]

A clear advantage of tagging all paraphrases in terms of recall and MRR is that we can determine the **upper bound** for the performance by selecting the highest rated item per ranking. In other words, we can imagine a system or an oracle that always picks one of the best options (see table 1). Hence, the upper bounds for MRR and recall are 0.417 and 0.309, respectively. Certainly, this is the highest performance any configuration or system can achieve operating on our two collections. Analogously, the **lower bound** for the performance is computed by singling out the lowest scored element in each ranking. For our corpus, the lower bound for MRR is 0.0004, whereas for recall it is 0.0073.

Moreover, our collections offer another reference for the performance. The title question (source paraphrase) yields a rough approximation of what a *human user* would prompt to a cQA service (cf. table 1). Remember that the title sets the discussion topic of a Yahoo! Answers page, and it is thus the reference read and clicked by the users of the search engine. By inspecting the performance accomplished by these titles, we obtain for our corpus: MRR=0.126 and recall=0.180.

We used two **baseline** methods for comparison. The **first baseline** (called BoW(G)) is built on top of the learning to rank SVMRank approach trained solely with BoW features. This vector space model is general in the sense that it is derived from all the examples embodied in the respective collection. We tuned its performance for several thresholds (word frequency counts from 0 to 19). In both cases (MRR and recall), the optimal threshold was 2, obtaining a performance of MRR=0.100 and recall=0.157. Normally, the BoW model supplies good performance in many

[6] Using http://secondstring.sourceforge.net/

[7] From now on, all MRR and recall values refer to the average values obtained when carrying out the cross-validation.

	Centroid Vector					
	Cosine	Manhattan	Euclidean	Squared Cord	Xi Squared	Canberra
Recall	0.154	0.127	0.142	0.144	0.147	0.123
MRR	0.094	0.076	0.091	0.094	0.099	0.059
	SVMRank		Corpus Statistics			
	BoW(G)	GFO(G)	Upper Bound	Yahoo! Titles	Lower Bound	
Recall	0.157	**0.164**	0.309	0.180	0.0073	
MRR	0.100	**0.109**	0.417	0.126	0.0004	

Table 3: Global corpus statistics and results obtained by our different baseline configurations (general models).

text mining applications. In our task, however, it only reached 23.96% of the achievable MRR and 50.79% of the achievable recall, respectively. This result is also below the potential human performance.

This **first baseline** is extended by means of a greedy algorithm for performing feature selection. It starts with an empty bag of features and after each iteration adds the one that performs the best. In order to determine this feature, the algorithm tests each non-selected property in conjunction with all the features in the bag. The procedure halts when there is no non-selected feature that enhances the performance. We refer to the system utilizing the best set of properties discovered by this algorithm as GFO(G). This greedy feature optimization (GFO) finished with the best baseline performance, that is with 0.164 and 0.109 for recall and MRR, respectively. In percentages, this translates into 53.07% (recall) and 26.14% (MRR) of the upper bounds. These values indicate a noticeable increase with respect to the BoW(G) models, underlining the usefulness of our battery of features listed in section 3.4.

For the **second baseline**, we utilized a centroid vector trained and tested via five-fold cross-validation. We used the same splits of our MRR and recall collections as used by our SVMRank general models. The vector is composed of terms that appear in at least three paraphrases, where each term is represented by the average MRR/recall values determined from the retrieved paraphrases. We tested six different measures to compute the similarity and distance to the centroid (see [33] for details on these metrics). Table 3 displays the results of the best scores reaped by this baseline: 0.099 (MRR) and 0.154 (recall). Note that the former is accomplished by benefiting from the Xi Squared distance metric, whereas the latter from the cosine similarity.

It is worth noting that all baselines are "*general models*" as they exploit the respective entire set of examples, contrary to specific models, which profit exclusively from the data belonging to the respective categories. Even though, GFO(G) improves the best performance by 4.46% on the recall collection, and by 9% on the MRR collection. However, none of these baseline systems outperform our human reference performance. In the following, we use the same empirical procedure to study the performance of the category-specific models for different levels of granularity.

4.1. First-level Categories

In our first analysis, we divide each collection into 26 different splits according to the first-level categories of Yahoo! Answers and selected by the asker when submitting the question to Yahoo! Answers. It is worth stressing that some questions, and thus the rankings they are in, might fall into several categories as they have been asked

Category Name	NoR	UB	Y!T	LB	GFO(G)	BoW(QC1)	GFO(QC1)
Arts & Humanities	1,926	0.217	0.124	0.005	0.112 (51.72)	0.129 (59.74)	**0.151** (69.52)
Beauty & Style	3,967	0.267	0.151	0.005	0.133 (49.80)	0.136 (51.19)	**0.159** (59.63)
Business & Finance	1,152	0.307	0.178	0.004	0.153 (49.93)	0.193 (62.90)	**0.222** (72.38)
Cars & Transportation	2,325	0.361	0.211	0.007	0.197 (54.44)	0.216 (59.73)	**0.253** (70.02)
Computers & Internet	1,669	0.349	0.194	0.006	0.174 (49.91)	0.194 (55.53)	**0.222** (63.64)
Consumer Electronics	1,138	0.396	0.222	0.011	0.208 (52.54)	0.236 (59.71)	**0.275** (69.39)
Dining Out	474	0.343	0.210	0.015	0.201 (58.52)	0.220 (64.09)	**0.252** (73.61)
Education & Reference	2,910	0.306	0.155	0.005	0.166 (54.16)	0.191 (62.55)	**0.213** (69.73)
Entertainment & Music	6,403	0.279	0.157	0.012	0.147 (52.89)	0.155 (55.58)	**0.170** (60.90)
Environment	296	0.222	0.125	0.003	0.090 (40.50)	0.152 (68.32)	**0.170** (76.65)
Family & Relationships	4,312	0.177	0.101	0.003	0.084 (47.68)	0.087 (49.33)	**0.101** (57.33)
Food & Drink	3,146	0.327	0.187	0.008	0.166 (50.81)	0.181 (55.24)	**0.209** (63.80)
Games & Recreation	1,365	0.353	0.199	0.010	0.190 (53.91)	0.216 (61.31)	**0.243** (68.96)
Health	4,050	0.247	0.141	0.003	0.121 (49.01)	0.134 (54.02)	**0.154** (62.32)
Home & Garden	1,781	0.328	0.180	0.003	0.165 (50.21)	0.189 (57.68)	**0.219** (66.80)
Local Businesses	115	0.267	0.170	0.001	0.138 (51.81)	0.207 (77.60)	**0.245** (91.87)
News & Events	487	0.184	0.103	0.003	0.087 (47.20)	0.121 (65.65)	**0.136** (73.77)
Pets	3,162	0.241	0.143	0.004	0.119 (49.61)	0.126 (52.29)	**0.144** (59.80)
Politics & Government	2,767	0.251	0.141	0.005	0.123 (49.13)	0.143 (57.20)	**0.165** (65.86)
Pregnancy & Parenting	4,671	0.231	0.138	0.006	0.128 (55.31)	0.134 (57.99)	**0.150** (64.87)
Science & Mathematics	4,087	0.359	0.207	0.007	0.181 (50.34)	0.202 (56.28)	**0.225** (62.77)
Social Science	1501	0.160	0.103	0.003	0.083 (51.60)	0.092 (57.47)	**0.108** (67.40)
Society & Culture	6,579	0.198	0.114	0.004	0.101 (51.16)	0.105 (53.12)	**0.118** (59.59)
Sports	3,744	0.332	0.201	0.015	0.188 (56.70)	0.201 (60.72)	**0.228** (68.81)
Travel	1,945	0.315	0.181	0.007	0.167 (52.94)	0.189 (59.77)	**0.217** (68.89)
Yahoo! Products	497	0.218	0.115	0.002	0.097 (44.48)	0.128 (58.89)	**0.140** (64.40)

Table 4: Results obtained for each first-level category in the recall collection. The table shows the respective corpus statistics. NoR stands for Number of Rankings, UB for Upper Bound, Y!T for Yahoo! Titles, and LB stands for Lower Bound. GFO(G) and GFO(QC1) stand for the figures obtained by performing feature optimization for the general and first-level category-specific models, respectively. BoW(QC1) represents the first-level category-specific model considering only words as features. The respective percentages of the upper bounds are given in parentheses.

multiple times, but categorized differently due to distinct interpretations. Since each of the new 52 categorized datasets is a subset of its respective MRR/recall collection, corpus statistics (i.e., upper and lower bounds together with the human reference) must be re-computed. Tables 4 and 5 show the results of the re-computations together with the figures achieved by the GFO(G) baseline when considering only its results for the rankings of the respective category. Analogously, BoW(QC1) and GFO(QC1) denote the outcomes accomplished by first-level category-specific models constructed on top of the bag-of-words feature and the array of attributes determined by GFO, respectively. The results reaped by BoW(QC1) and GFO(QC1) were obtained via 5-fold cross-validation operating on the split corresponding to the category. From these figures, it is worth pointing out the following findings:

1. Our recall collection consists mainly of questions extracted from the categories: "*Society & Culture*" and "*Entertainment & Music*"; while the MRR collection is composed mainly of elements derived from the categories: "*Health*" and "*Science & Mathematics*". Interestingly enough, these last two MRR categories are known to bear more informational than subjective questions [2], and their relatively high upper bounds signal that effec-

Category Name	NoR	UB	Y!T	LB	GFO(G)	BoW(QC1)	GFO(QC1)
Arts & Humanities	1,138	0.351	0.108	0.001	0.091 (25.94)	0.149 (42.47)	**0.194** (55.14)
Beauty & Style	2,442	0.368	0.110	0.001	0.092 (24.88)	0.118 (31.92)	**0.156** (42.30)
Business & Finance	2,651	0.463	0.144	0.001	0.123 (26.67)	0.161 (34.75)	**0.208** (45.00)
Cars & Transportation	4,410	0.398	0.103	0.001	0.093 (23.42)	0.122 (30.59)	**0.159** (39.90)
Computers & Internet	3,498	0.389	0.117	0.001	0.100 (25.69)	0.126 (32.40)	**0.164** (42.20)
Consumer Electronics	3,547	0.412	0.114	0.001	0.104 (25.32)	0.122 (29.63)	**0.164** (39.81)
Dining Out	297	0.410	0.100	0.001	0.102 (24.78)	0.176 (42.89)	**0.260** (63.37)
Education & Reference	2,834	0.392	0.127	0.001	0.116 (29.60)	0.145 (37.11)	**0.183** (46.61)
Entertainment & Music	2,954	0.363	0.107	0.001	0.098 (26.86)	0.137 (37.61)	**0.176** (48.34)
Environment	154	0.333	0.084	0.001	0.055 (16.49)	0.190 (57.10)	**0.242** (72.82)
Family & Relationships	1,124	0.319	0.096	0.001	0.063 (19.77)	0.107 (33.58)	**0.149** (46.65)
Food & Drink	2,660	0.420	0.121	0.001	0.100 (23.84)	0.134 (31.88)	**0.184** (43.69)
Games & Recreation	2,520	0.413	0.108	0.001	0.097 (23.59)	0.149 (36.12)	**0.203** (49.31)
Health	5,182	0.422	0.136	0.001	0.107 (25.47)	0.131 (31.09)	**0.169** (40.00)
Home & Garden	3,119	0.420	0.099	0.001	0.094 (22.27)	0.127 (30.21)	**0.167** (39.66)
Local Businesses	352	0.469	0.131	0.001	0.131 (28.01)	0.250 (53.33)	**0.301** (66.10)
News & Events	220	0.342	0.107	0.001	0.087 (25.60)	0.192 (56.26)	**0.241** (70.56)
Pets	1,648	0.383	0.102	0.001	0.079 (20.68)	0.112 (29.30)	**0.155** (40.60)
Politics & Government	2,235	0.416	0.121	0.001	0.104 (24.96)	0.150 (35.98)	**0.202** (48.48)
Pregnancy & Parenting	1,971	0.323	0.087	0.001	0.072 (22.29)	0.104 (32.08)	**0.137** (42.42)
Science & Mathematics	5,044	0.449	0.157	0.001	0.131 (29.13)	0.152 (33.85)	**0.187** (41.73)
Social Science	726	0.386	0.156	0.001	0.129 (33.37)	0.183 (47.38)	**0.218** (56.44)
Society & Culture	2,084	0.376	0.130	0.001	0.104 (27.82)	0.141 (37.65)	**0.180** (48.03)
Sports	2,392	0.423	0.112	0.001	0.112 (26.51)	0.159 (37.62)	**0.211** (49.98)
Travel	2,231	0.487	0.150	0.001	0.135 (27.79)	0.183 (37.54)	**0.256** (52.63)
Yahoo! Products	690	0.377	0.122	0.001	0.090 (24.00)	0.131 (34.68)	**0.178** (47.32)

Table 5: Results obtained for each 1st level category in the MRR collection. The table shows the respective corpus statistics. NoR stands for Number of Rankings, UB for Upper Bound, Y!T for Yahoo! Titles, and LB stands for Lower Bound. GFO(G) and GFO(QC1) stand for the figures obtained by performing feature optimization for the general and first-level category-specific models, respectively. BoW(QC1) represents the first-level category-specific model considering only words as features. The respective percentages of the upper bounds are given in parentheses.

tive paraphrases were found across search logs. In a similar manner, a comparatively large amount of good paraphrases were also acquired for "*Science & Mathematics*" within the recall collection.

2. Analyzing the GFO(G) model operating on the different categories, we discover that it performs better on "*Dining Out*" (58.52% of the achievable recall) and "*Social Science*" (33.33% of the MRR upper bound), while in both cases, its worst performance is on the "*Environment*" category. The differences between both extremes are 18.02% and 16.88% of the potential recall and MRR, respectively. Particularly, in the MRR collection, the performance for "*Social Science*" almost doubles the performance for "*Environment*" (16.49% of the upper bound). Only for the MRR category "*Dining Out*" and for the recall category "*Education & Reference*", GFO(G) performs better than our human reference.
3. In light of the fact that BoW(QC1) outperformed GFO(G) in all categories and collections, we can conclude that different and category-specific word distributions are observed across effective paraphrases. As a natural consequence, our results indicate that first-level category-specific models, grounded on a simple bag-of-words vector

space, are a cost-efficient solution as the extraction of extra features demand extra computational resources. This conclusion holds for MRR and recall indistinctly, and it is valid for large and small categories. For instance, major improvements can be found in small MRR categories (see table 5): "*Environment*" 40.61% (154 items), "*News & Events*" 30.66% (220 items) and "*Local Business*" 25.32% (352 items). Larger categories also experienced significant improvements getting closer to their potential upper bounds: "*Health*" 5.62% (5,182 items), "*Science & Mathematics*" 4.71% (5,044 items) and "*Cars & Transportation*" 7.18% (4,410 items).

4. Let A be the inverse of the number of samples, and B be the increase in terms of the percentage of the upper bound achieved by BoW(QC1) with respect to GFO(G). The Pearson correlation coefficient between A and B across the 26 categories supports the finding that category-specific models better capture word distribution patterns observed for small categories, ergo enhancing their performance: 0.95 (MRR) and 0.80 (recall). Both numbers denote a strong correlation.
5. Further, GFO(QC1) represents the results obtained by exploiting our greedy feature selection algorithm. Note that GFO(QC1) led to marked improvements for the recall categories: "*Local Business*" 14.27% (115 samples) and "*Cars & Transportation*" 10.29% (2,325 samples); while producing relatively modest growths for the category "*Yahoo! Products*" 5.51% (497 samples). The increase of 14.27% "*Local Business*" means that it now reaches 91.87% of the upper bound, which is a comparatively high performance. Similarly, GFO(QC1) obtained substantial increases for MRR categories: "*Dining Out*" 20.48% (297 samples) and "*Environment*" 15.72% (154 samples); it brought about relatively minor enhancements for the MRR categories "*Health*" 8.91% (5,182 samples) and "*Cars & Transportation*" 9.31% (4,410 samples). The Pearson coefficient points out to the fact that feature optimization reduces the dependence of the performance on the amount of examples for the categories: 0.64 (MRR) and 0.63 (recall). In light of this outcome, we conclude that GFO(QC1) helps to tackle data-sparseness by drawing more effective generalizations, i.e., it is able to learn category-specific attributes more effectively.
6. As for the most salient properties, the first three attributes selected by GFO were unigrams, bigrams and trigrams. In each collection, this triplet of properties was chosen for 24 out of the 26 categories. This indicates that different word distributions are found across distinct categories, and it also provides a good starting point for a comparison based exclusively on lexical features. With regard to other features, adverbial and quantifier collocations are prominent across GFO(QC1) models for both recall and MRR collections; the number of NNS, RBS, RBR were also recurrently chosen across GFO(QC1) models for both collections; word lemma was incorporated into six and nine GFO(QC1) models for the MRR and recall collection, respectively; concerning the string similarity measures, the JACCARD distance was selected for six GFO(QC1) recall models.
7. Using the discriminative phrases listed in [2], we roughly estimated the fraction of objective and subjective questions for each category. We computed the Pearson coefficient between the number of both classes of question intents across the 26 categories. This coefficient is -0.52, indicating a strong anti-correlation. This means that when the fraction of question intents of one class (objective or subjective) is high for a category, the value

for the other class is likely to be low. Some interesting examples are the categories (objective-subjective): "*Computer & Internet*" (16.16%-4.72%), "*Social Science*" (12.04%-7.02%), "*Travel*" (10.99%-9.88%) , "*Pets*" (9.72%-4.44%), "*Family & Relationships*" (9.47%-10.28%), "*Sports*" (4.28%-19.80%), "*Education & Reference*" (3.40%-19.69%), "*Cars & Transportation*" (4.33%-18.88%), "*Yahoo! Products*" (5.73%-18.36%).

8. Let *A* be the percentage of objective or subjective questions for a category, and *B* be the increase in performance of GFO(QC1) over BoW(QC1) for a category. The Pearson coefficient between *A* and *B* across the 26 categories is: objective-MRR 0.036, objective-recall -0.15, subjective-MRR 0.06, and subjective-recall 0.019. If *B* is the increase of GFO(QC1) over GFO(G) for a category, we obtain: objective-MRR 0.13, objective-recall -0.12, subjective-MRR -0.14, and subjective-recall 0.019.

   A stronger correlation between objective/subjective questions and corresponding improvements is observed in the event of GFO(G) instead of BoW(QC1), which indicates that after data-splitting, models become less sensitive to the question intent, especially when dealing with the MRR collection. In other words, our category-specific models adapt well to the degree of objectivity and subjectivity of each category, because improvements are less connected to a particular question intent.

In summary, first-level category-specific models are more fitted to recognize effective paraphrases in cQA than general models, independently on whether we want to enhance retrieval (recall) or ranking (MRR). Our analysis shows that one key reason behind this greater suitability is that category-specific models adapt better to the degree of objectivity and subjectivity of each particular category, especially by modeling specific word distribution patterns. In addition, our figures also indicate that a simple category-based BoW strategy is a cost-efficient solution as it clearly outperforms general models enriched with assorted features. Along the same line, our results reveal that unigrams, bigrams and trigrams are key features to model the specificities of each category.

4.2. Second-level Categories

Following an analogous approach, we examined the impact of second-level categories on the performance. On the one hand, second-level categories are more fine-grained than first-level categories, but on the other hand, they usually contain a smaller number of samples. For these reasons, we considered only categories with more than 100 rankings in our analysis. This means that we studied 150 (MRR) and 160 (recall) second-level categories. Tables 6 and 7 display some interesting results. BoW(QC2) denotes the model built from the bag-of-word view of the elements belonging to the respective second-level category while GFO(QC2) refers to the model constructed from the view generated with the features determined by GFO. From these experiments, it is worth noting:

1. In 71 out of the 150 MRR categories, BoW(QC2) outperformed GFO(QC1), leading to an overall average improvement of 0.087% of the upper bound. Likewise, for recall, we found that the average increase accomplished by BoW(QC2) over GFO(QC1) was 0.44% of the upper bound, improving the performance in 83 out of 160 cases. All in all, these outcomes corroborate the finding that a fine-grained categorized bag-of-words model is a

Category Name	NoR	UB	Y!T	LB	GFO(G)	GFO(QC1)	BoW(QC2)	GFO(QC2)
Entertainment & Music→Polls & Surveys	4,015	0.227	0.132	0.011	0.121 (53.30)	0.142 (62.37)	0.130 (57.03)	**0.149** (65.46)
Society & Culture→Religion & Spirituality	3,102	0.167	0.100	0.003	0.089 (52.83)	0.097 (57.98)	0.091 (54.82)	**0.105** (62.75)
Family & Relationships→Singles & Dating	2,555	0.166	0.096	0.003	0.081 (48.87)	0.093 (55.93)	0.082 (49.61)	**0.097** (58.37)
Society & Culture→Cultures & Groups	2,027	0.186	0.107	0.004	0.091 (48.93)	0.104 (56.11)	0.096 (51.57)	**0.108** (58.32)
Pets→Dogs	1,536	0.233	0.139	0.003	0.114 (49.04)	0.136 (58.30)	0.120 (51.62)	**0.137** (58.79)
Beauty & Style→Fashion & Accessories	1,415	0.265	0.144	0.006	0.134 (50.66)	0.156 (58.97)	0.149 (56.16)	**0.169** (63.83)
Pregnancy & Parenting→Baby Names	1,409	0.323	0.204	0.013	0.211 (65.30)	0.231 (71.60)	0.235 (72.77)	**0.247** (76.58)
Education & Reference→Words & Wordplay	1,323	0.312	0.146	0.005	0.181 (58.18)	0.224 (71.82)	0.212 (67.84)	**0.230** (73.57)
Beauty & Style→Other - Beauty & Style	1,283	0.217	0.124	0.006	0.110 (50.94)	0.129 (59.53)	0.125 (57.61)	**0.144** (66.68)
Pregnancy & Parenting→Pregnancy	1,212	0.197	0.108	0.001	0.090 (45.44)	0.117 (59.19)	0.104 (52.74)	**0.124** (62.73)
Dining Out→Fast Food	278	0.383	0.245	0.021	0.237 (61.94)	**0.297** (77.60)	0.248 (64.63)	0.276 (72.14)
Sports→Swimming & Diving	117	0.267	0.163	0.006	0.130 (48.63)	0.173 (64.55)	0.188 (70.33)	**0.219** (81.86)
Sports→Golf	115	0.329	0.206	0.008	0.177 (53.88)	0.242 (73.69)	0.220 (66.96)	**0.271** (82.36)
Travel→Travel (General)	113	0.283	0.150	0.006	0.132 (46.32)	0.188 (66.36)	0.214 (75.65)	**0.234** (82.59)
Business & Finance→Small Business	112	0.195	0.117	0.003	0.089 (45.52)	0.129 (65.98)	0.154 (78.81)	**0.176** (89.96)
Consumer Electronics→Other - Electronics	109	0.335	0.166	0.015	0.160 (47.94)	0.224 (67.00)	0.248 (74.00)	**0.280** (83.80)
Cars & Transportation→Aircraft	109	0.248	0.136	0.008	0.124 (50.03)	0.180 (72.85)	0.185 (74.53)	**0.202** (81.52)
Social Science→Economics	108	0.230	0.148	0.005	0.100 (43.56)	0.170 (73.83)	0.153 (66.58)	**0.183** (79.65)
Arts & Humanities→Poetry	106	0.173	0.103	0.001	0.065 (37.66)	0.112 (64.38)	0.108 (62.18)	**0.140** (80.74)
Arts & Humanities→Performing Arts	106	0.203	0.133	0.004	0.116 (57.19)	0.134 (66.22)	0.146 (72.10)	**0.169** (83.26)
Travel→United Kingdom	106	0.307	0.164	0.004	0.152 (49.62)	0.190 (61.83)	0.225 (73.37)	**0.252** (82.19)
Sports→Auto Racing	102	0.273	0.158	0.008	0.126 (46.23)	0.175 (64.06)	0.203 (74.27)	**0.213** (78.22)

Table 6: Results obtained for each second-level category in the recall collection. It shows the respective corpus statistics. NoR stands for Number of Rankings, UB for Upper Bound, Y!T for Yahoo! Titles, and LB stands for Lower Bound. GFO(G) and GFO(QC1) stand for the figures obtained when performing feature optimization for the general and first-level category-specific models, respectively. BoW(QC2) and GFO(QC2) represent the respective second-level category-specific models. In parentheses, the respective percentage of the upper bound.

better cost-efficient solution than its respective "*father*" general model build on top of more complex semantic and syntactic structures. This is due to the fact that these models capture effective patterns of word distributions that are specific for each category.

2. Conversely, GFO(QC2) outperformed GFO(QC1) in all but two MRR categories. The average enhancement was 14.36% of the achievable MRR. Similarly, GFO(QC2) outperformed GFO(QC1) in all but one category (see table 6), finishing with an average growth of 10.37% of the achievable recall.

   Let A be the inverse of the number of samples, and B be the increase in terms of the percentage of the upper bound achieved by BoW(QC2) with respect to GFO(QC1). The Pearson coefficient between A and B is 0.43. Applying the same computation to GFO(QC2) and GFO(QC1), we obtain a value of 0.55. This stronger correlation signifies that GFO(QC2) has a greater impact on smaller categories. Thus it mitigates data-sparseness by inferring more effective abstractions from the data, indicating not only that our feature set is useful, but also that it is possible to learn category-specific attributes more efficiently. For instance, tables 6 and 7 show the results for the ten largest and eight smallest second-level categories for both collections. For smaller categories, we observe models reaching over 78% of the achievable recall and over 65% of the achievable MRR, which in average is notoriously better compared to larger categories.

Category Name	NoR	UB	Y!T	LB	GFO(G)	GFO(QC1)	BoW(QC2)	GFO(QC2)
Cars & Transportation→Car Makes	1,762	0.390	0.101	0.0004	0.099 (25.29)	0.163 (41.79)	0.130 (33.32)	**0.171** (43.84)
Games & Recreation→Video & Online Games	1,734	0.405	0.098	0.0005	0.089 (21.96)	0.191 (47.19)	0.149 (36.70)	**0.200** (49.35)
Health→Diseases & Conditions	1,572	0.437	0.147	0.0003	0.114 (26.12)	0.179 (40.88)	0.156 (35.76)	**0.210** (47.97)
Cars & Transportation→Maintenance & Repairs	1,337	0.369	0.083	0.0002	0.076 (20.50)	0.124 (33.56)	0.114 (30.93)	**0.165** (44.69)
Consumer Electronics→Cell Phones & Plans	1,279	0.404	0.122	0.0004	0.110 (27.34)	0.150 (37.22)	0.139 (34.52)	**0.174** (43.00)
Food & Drink→Cooking & Recipes	1,197	0.406	0.101	0.0003	0.092 (22.58)	0.173 (42.49)	0.141 (34.75)	**0.183** (45.10)
Computers & Internet→Hardware	1,142	0.367	0.099	0.0003	0.091 (24.66)	0.149 (40.62)	0.135 (36.75)	**0.173** (47.22)
Science & Mathematics→Biology	1,116	0.426	0.141	0.0005	0.124 (29.04)	0.172 (40.44)	0.160 (37.60)	**0.201** (47.16)
Science & Mathematics→Chemistry	1,108	0.479	0.162	0.0004	0.143 (29.92)	0.192 (40.10)	0.196 (40.96)	**0.241** (50.36)
Home & Garden→Maintenance & Repairs	962	0.423	0.086	0.0002	0.100 (23.56)	0.162 (38.31)	0.160 (37.95)	**0.224** (52.98)
Yahoo! Products→Yahoo! Mail	334	0.393	0.139	0.0004	0.107 (27.27)	**0.185** (47.12)	0.121 (30.71)	0.181 (46.11)
Dining Out→Fast Food	152	0.388	0.087	0.0004	0.080 (20.73)	**0.254** (65.49)	0.199 (51.54)	0.243 (62.76)
Health→Optical	117	0.429	0.104	0.0006	0.113 (26.22)	0.187 (43.54)	0.205 (47.65)	**0.284** (66.24)
Cars & Transportation→Boats & Boating	109	0.443	0.111	0.0001	0.121 (27.24)	0.164 (37.09)	0.211 (47.62)	**0.291** (65.85)
Pregnancy & Parenting→Other	107	0.260	0.047	0.0000	0.041 (15.75)	0.105 (40.49)	0.123 (47.39)	**0.175** (67.38)
Cars & Transportation→Safety	105	0.506	0.144	0.0003	0.151 (29.92)	0.200 (39.48)	0.303 (59.83)	**0.355** (70.21)
News & Events→Current Events	104	0.294	0.089	0.0005	0.087 (29.45)	0.207 (70.25)	0.179 (60.76)	**0.220** (74.73)
Games & Recreation→Toys	103	0.397	0.135	0.0003	0.130 (32.80)	0.189 (47.74)	0.217 (54.79)	**0.272** (68.54)
Consumer Electronics→Cameras	103	0.396	0.048	0.0002	0.082 (20.62)	0.150 (37.96)	0.210 (52.99)	**0.268** (67.61)
Education & Reference→Teaching	102	0.370	0.105	0.0002	0.081 (21.83)	0.154 (41.60)	0.180 (48.58)	**0.253** (68.39)

Table 7: Results obtained for each 2nd level category in the MRR collection. It shows the respective corpus statistics. GFO(G) and GFO(QC1) stand for the figures obtained when performing feature optimization for the general and first-level category-specific models, respectively. BoW(QC2) and GFO(QC2) represent the respective second-level category-specific models. In parentheses, the respective percentage of the upper bound.

3. In terms of objectivity versus subjectivity, if we account for the difference in terms of the percentage of the upper bound achieved by GFO(QC2) and GFO(QC1), we also find consistent improvements across both collections and question intents. Take for instance the sub-categories derived from the three first-level categories that bear a larger portion of subjective questions (e.g., "*Yahoo! Products*"). In these sub-categories, the performance gets closer to the upper bound by an average of 9.51% (recall). Performing the same analysis for descendants of the three first-level categories that embody a larger fraction of objective questions (e.g., "*Computer & Internet*"), the performance gets closer to the upper bound by an average of 9.17% (recall). Concerning MRR, subjective questions obtain 14.74% while objective questions obtain 13.18%. In light of these figures, we can conclude that GFO(QC2) adapts better than GFO(QC1) to the different degrees of objectivity and subjectivity of the second-level categories.

Overall, our experiments point out to the fact that second-level category-specific are more effective than first-level models in terms of improving retrieval and ranking. They adapt even better to the degree of objectivity/subjectivity of each category, and they can make better use of attributes for tackling data-sparseness head-on.

4.3. Third-level Categories and Question-types

We studied the effect of third-level categories on the performance in an analogous way. Due to the fact that most of these categories are very small, we concentrated our analysis on the five largest units. The outcomes are displayed

Category Name	NoR	UB	Y!T	LB	GFO(G)	GFO(QC1)	GFO(QC2)	BoW(QC3)	GFO(QC3)
MRR									
Computers & Internet→Hardware→...									
...Laptops & Notebooks	455	0.363	0.102	0.0002	0.105 (29.03)	0.143 (39.00)	0.173 (47.62)	0.134 (36.95)	**0.185** (50.81)
Beauty & Style→Skin & Body→Other	444	0.33	0.094	0.0003	0.081 (24.58)	0.141 (42.58)	0.172 (52.04)	0.137 (41.64)	**0.187** (56.63)
Business & Finance→Taxes→United States	369	0.472	0.127	0.0002	0.111 (23.53)	0.170 (36.03)	0.198 (41.99)	0.157 (33.16)	**0.228** (48.32)
Health→ Diseases & Conditions→Other	367	0.415	0.125	0.0002	0.112 (27.01)	0.146 (35.29)	0.199 (48.02)	0.178 (42.89)	**0.248** (59.66)
Sports→Outdoor Recreation→Hunting	365	0.441	0.098	0.0002	0.089 (20.16)	0.197 (44.74)	0.221 (50.08)	0.158 (35.89)	**0.231** (52.28)
Recall									
Society & Culture→Cultures & Groups→...									
...Other - Cultures & Groups	1,252	0.191	0.107	0.004	0.091 (47.75)	0.100 (52.27)	0.103 (53.88)	0.097 (50.90)	**0.116** (60.55)
...Lesbian. Gay. Bisexual. and Transgendered	912	0.158	0.094	0.003	0.080 (50.54)	0.092 (58.05)	0.090 (57.08)	0.094 (59.59)	**0.109** (68.90)
Beauty & Style →Skin & Body→Other	655	0.230	0.134	0.005	0.115 (49.94)	0.139 (60.25)	0.153 (66.24)	0.135 (58.42)	**0.160** (69.27)
Society & Culture →Holidays→Ramadan	520	0.155	0.093	0.003	0.080 (51.39)	0.091 (58.94)	0.106 (68.27)	0.093 (60.11)	**0.112** (72.30)
Entertainment & Music→Music→Rock'n'Pop	385	0.28	0.165	0.022	0.152 (53.63)	0.171 (60.32)	0.200 (70.81)	0.178 (62.91)	**0.217** (76.52)

Table 8: Results obtained for the five largest third-level categories (MRR and recall collections). In parentheses, the respective percentage of the upper bound.

in table 8. In summary, the results revealed the same trend that our earlier experiments targeted at broader categories. Basically, our results indicate that GFO(CQ3) outperforms all other configurations regardless the collection. Our outcomes also show that GFO(CQ2) performs better than GFO(CQ1) in all but one case; GFO(CQ1) is better than GFO(G) in all cases. This ratifies that fine-grained categorized models are a more effective solution to recognize useful paraphrases for answer ranking and retrieval from cQA archives. Although, BoW(CQ3) is not a major competitor for GFO(CQ2) – unlike BoW(CQ2) for GFO(CQ1) and BoW(CQ1) for GFO(G) – it is still a better alternative than GFO(G), signifying that models based on categorized bag-of-words yield cost-efficient solutions. The significant increase in performance between GFO(CQ3) and GFO(CQ2), and between GFO(CQ3) and BoW(CQ3) corroborates that our feature set is helpful to infer good generalizations of the underlying semantic and syntactic structures of each category.

Finally, we examined the effect of question-types on category-specific models. More specifically, we extended the third-level of the Yahoo! Answers question taxonomy with Wh-question typification. For the two largest subsets in table 8, we identified the type of question embodied in each ranking by checking whether its posted question starts with any of the syntactic patterns provided by an external taxonomy, cf. [34]. If no pattern matched, we marked the question as "*unmatched*". Note that patterns were solely aligned with the posted question, not the paraphrases distilled from search logs. This strategy recognized only one prominent type within each data-set, viz. *why* (recall) and *how-procedural* (MRR).

For both largest third-level categories, the figures in table 9 indicate that adding question typification brings about a significant increase from 50.81% to 58.50% of the MRR upper bound; while from 60.55% to 62.26% of the achievable upper bound for recall. Adding question typification to the BoW approach also showed an improvement of about 4% for recall and about 8% for MRR. These results are quite consistent with the results reported in our previous work

Recall Category: Society & Culture→Cultures & Groups→Other - Cultures & Groups				MRR Category: Computers & Internet→Hardware→Laptops & Notebooks			
Question Type	No. Rankings	BoW(QC3T)	GFO(QC3T)	Question Type	No. Rankings	BoW(QC3T)	GFO(QC3T)
why-qids	330	0.076	0.090	how-procedural	80	0.099	0.159
unmatched	922	0.115	0.130	unmatched	375	0.176	0.224
Total	1,252	0.105	**0.119**	Total	455	0.162	**0.212**
	% of Upper Bound	54.76	**62.26**		% of Upper Bound	44.71	**58.50**

Table 9: Results obtained for the largest third-level category enriched with question types (recall and MRR).

(cf. [15]), showing that the Yahoo! Answers category system can be automatically enlarged with question types to improve the performance of some text mining tasks.

5. Conclusions

In recent years, cQA platforms have become a viable alternative to get answers to our questions by asking other members of a community. The advantage of cQA systems lies on the fact that answers to some questions cannot or are hard to be found directly within web documents. In particular, many times the generation of these answers requires the synthesis of facts, experiences and world knowledge of the members of the community.

One of the problems with this interaction regards the fact that many times users enter bad formulations of their questions, inadvertently. This not only increases the chances of these questions to go unresolved, but also establishing their relations to past questions becomes more difficult. Taking advantage of answers in the archives is key to make the cQA system more vibrant, because this reduces the time delay between the posting of the new question and the submission of the corresponding good answers by other community members. Our work aims at bridging this gap by automatically detecting effective paraphrases for questions prompted by community members. These paraphrases can offer more effective suggestions to the user, for example, in searching the archives for past answers. In the same vein, cQA platforms can benefit from these effective paraphrases for internally searching the archives, locating good past answers and potential experts, which can lead to a reduction of the answering delay.

In short, we designed a framework to study the effectiveness of paraphrases based on a massive data-set of automatically rated question paraphrases. This collection was acquired by exploiting the connections between Yahoo! Answers and Yahoo! Search logs. Basically, we tested broad and fine-grained models that take into account the categorization provided by the members at the time of submitting their questions. Our study reveals that there is strong relation between categories and the subjective/objective intent of their questions, which substantially impacts on the detection of effective paraphrases. In order to deal with this, we built category-specific learning to rank models (i.e., SVMRank), showing that these can adapt well to the different degrees of objectivity and subjectivity of each category, since improvements are less correlated to question intents. More precisely, we examined category-specific models of different granularity levels. In so doing, we made allowances for the three levels supplied by the Yahoo! Answers question categorization system.

Fine-grained category models rely on large-scale data for inferring category-specific feature distributions. However, such an amount of training material is not always available. Nevertheless, the level of granularity can be tuned in tandem with the amount of available data, so that data-sparseness does not hurt the performance of the models. In addition, the linguistics of web queries is also an obstacle to explore a wider variety of ranking features as most NLP toolkits are designed for dealing with documents, not with search queries. Nonetheless, it is possible to train purpose-built tools for search queries, whereby linguistic-oriented attributes can be extracted, and hence incorporating them into the ranking models. At any rate, this kind of solution is very demanding, since it requires major efforts into model design and experimentation.

In summary, our experiments reveal that the more specific the models are the better are the results. Incidentally, the features that have been shown to be most effective when they are integrated into the best ranking models include unigrams, bigrams and trigrams. Further, some types of collocations, syntactic categories and semantic relations have proven to be instrumental, and morphological analysis also have shown to be effectual.

Given these findings, we extended the question category system with Wh-keyword question typification, showing that this syntactic information is also promising for enhancing the detection of effective paraphrases. We also envisage the interpolation of general and category-specific models of different granularity levels as a means of enhancing the ranking system. Note that improving the performance of such a system does not only help answer retrieval and ranking, but also question routing, since finding good past answers can cooperate on assigning new questions to suitable experts.

On a final note, our results can also contribute to the subject of paraphrase generation. More specifically, our findings suggest that category-specific approaches might also be a better alternative to general models for producing effective paraphrases, since these strategies also need to consider different degrees of objectivity and subjectivity across categories. Moreover, our results also suggest that paraphrase generation techniques might need to take into account the incorporation of effective attributes into their models in consonance with the categories.

6. Acknowledgements

This work was partially supported by the projects FONDEF-IdeA (CA12I10081) and Fondecyt "*Bridging the Gap between Askers and Answers in Community Question Answering Services*" (11130094) funded by the Chilean Government, and the European Communitys Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 287923 (EXCITEMENT).

References

[1] A. Rechavi, S. Rafaeli, Knowledge and social networks in yahoo! answers, 2013 46th Hawaii International Conference on System Sciences 0 (2012) 781–789.

[2] F. M. Harper, D. Moy, J. A. Konstan, Facts or friends?: distinguishing informational and conversational questions in social q& a sites, in: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '09, ACM, New York, NY, USA, 2009, pp. 759–768.

[3] L. Chen, D. Zhang, L. Mark, Understanding user intent in community question answering, in: Proceedings of the 21st international conference companion on World Wide Web, WWW '12 Companion, ACM, New York, NY, USA, 2012, pp. 823–828.

[4] Y. Liu, S. Li, Y. Cao, C.-Y. Lin, D. Han, Y. Yu, Understanding and Summarizing Answers in Community-Based Question Answering Services, in: International Conference on Computational Linguistics, 2008, pp. 497–504.

[5] M. Liu, Y. Liu, Q. Yang, Predicting best answerers for new questions in community question answering, in: Web-Age Information Management, Springer, 2010, pp. 127–138.

[6] L. Yang, S. Bao, Q. Lin, X. Wu, D. Han, Z. Su, Y. Yu, Analyzing and predicting not-answered questions in community-based question answering services, in: AAAI, 2011.

[7] A. Shtok, G. Dror, Y. Maarek, I. Szpektor, Learning from the past: answering new questions with past answers, in: Proceedings of the 21st international conference on World Wide Web, WWW '12, ACM, New York, NY, USA, 2012, pp. 759–768.

[8] Y. Cao, H. Duan, C. Y. Lin, Y. Yu, H. W. Hon, Recommending questions using the mdl-based tree cut model, in: Proceedings of the 17th international conference on World Wide Web, WWW '08, ACM, 2008, pp. 81–90.

[9] J. Bian, Y. Liu, E. Agichtein, H. Zha, Finding the right facts in the crowd: factoid question answering over social media, in: World Wide Web Conference Series, 2008, pp. 467–476.

[10] M. J. Blooma, A. Y. K. Chua, D. H.-L. Goh, Quadripartite graph-based clustering of questions, in: Proceedings of the 2011 Eighth International Conference on Information Technology: New Generations, IEEE Computer Society, Washington, DC, USA, 2011, pp. 591–596.

[11] M. J. Blooma, J. C. Kurian, Clustering Similar Questions in Social Question Answering Systems, in: Proceedings of the 2012 Pacific Asia Conference on Information Systems (PACIS),, 2012.

[12] K. Wanga, Z. Ming, T. seng Chua, A syntactic tree matching approach to finding similar questions in community-based qa services, in: Research and Development in Information Retrieval, 2009, pp. 187–194.

[13] C.-Y. Lin, Automatic question generation from queries, in: Proceedings of Workshop on the Question Generation Shared Task and Evaluation Challenge, 2008, pp. 929–937.

[14] S. Zhao, H. Wang, C. Li, T. Liu, Y. Guan, Automatically generating questions from queries for community-based question answering, in: Proceedings of 5th International Joint Conference on Natural Language Processing, 2011, pp. 929–937.

[15] A. Figueroa, G. Neumann, Learning to Rank Effective Paraphrases from Query Logs for Community Question Answering, in: AAAI 2013, 2013.

[16] B. Li, Y. Liu, A. Ram, E. Garcia, E. Agichtein, Exploring question subjectivity prediction in community QA, Proceedings of the 31st annual international ACM SIGIR.

[17] O. Ferrandez, C. Spurk, M. Kouylekov, I. Dornescu, S. Ferrandez, M. Negri, R. Izquierdo, D. Tomas, C. Orasan, G. Neumann, et al., The qall-me framework: A specifiable-domain multilingual question answering architecture, Web semantics: Science, services and agents on the world wide web 9 (2) (2011) 137–145.

[18] X. Xue, J. Jeon, W. B. Croft, Retrieval models for question and answer archives, in: Research and Development in Information Retrieval, 2008, pp. 475–482.

[19] T. Joachims, Training Linear SVMs in Linear Time, in: Proceedings of the ACM Conference on Knowledge Discovery and Data Mining (KDD), 2006.

[20] B. M. John, A. Y.-K. Chua, D. H.-L. Goh, What makes a high-quality user-generated answer?, IEEE Internet Computing 15 (1) (2011) 66–71.

[21] J. Jeon, W. B. Croft, J. H. Lee, S. Park, A framework to predict the quality of answers with non-textual features, in: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval, SIGIR '06, ACM, New York, NY, USA, 2006, pp. 228–235.

[22] E. Agichtein, C. Castillo, D. Donato, A. Gionis, G. Mishne, Finding high-quality content in social media, in: Proceedings of the 2008 International Conference on Web Search and Data Mining, WSDM '08, ACM, New York, NY, USA, 2008, pp. 183–194.

[23] M. A. Suryanto, E. P. Lim, A. Sun, R. H. L. Chiang, Quality-aware collaborative question answering: methods and evaluation, in: Proceedings of the Second ACM International Conference on Web Search and Data Mining, WSDM '09, ACM, New York, NY, USA, 2009, pp. 142–151.

[24] F. M. Harper, D. Raban, S. Rafaeli, J. A. Konstan, Predictors of answer quality in online Q&A sites, in: Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems, CHI '08, ACM, New York, NY, USA, 2008, pp. 865–874.

[25] J. Atkinson, A. Figueroa, C. Andrade, Evolutionary optimization for ranking how-to questions based on user-generated contents, Expert Syst. Appl. 40 (17) (2013) 7060–7068.

[26] M. Surdeanu, M. Ciaramita, H. Zaragoza, Learning to rank answers on large online qa collections, in: In Proceedings of the 46th Annual Meeting for the Association for Computational Linguistics: Human Language Technologies (ACL-08: HLT, 2008, pp. 719–727.

[27] M. Surdeanu, M. Ciaramita, H. Zaragoza, Learning to rank answers to non-factoid questions from web collections, in: Computational Linguistics, Vol. 37, 2011, pp. 351–383.

[28] Z.-M. Zhou, M. Lan, Z.-Y. Niu, Y. Lu, Exploiting user profile information for answer ranking in cqa, in: Proceedings of the 21st international conference companion on World Wide Web, WWW '12 Companion, ACM, New York, NY, USA, 2012, pp. 767–774.

[29] D. E. Rose, D. Levinson, Understanding user goals in web search, in: Proceedings of the 13th international conference on World Wide Web, WWW '04, ACM, New York, NY, USA, 2004, pp. 13–19.

[30] S. Zhao, X. Lan, T. Liu, S. Li, Application-driven statistical paraphrase generation, in: Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2 - Volume 2, ACL '09, 2009, pp. 834–842.

[31] J.-R. Wen, J.-Y. Nie, H. Zhang, Query clustering using user logs, ACM Trans. Inf. Syst. 20 (1) (2002) 59–81.

[32] S. Zhao, H. Wang, T. Liu, Paraphrasing with search engine query logs, in: COLING 2010, 23rd International Conference on Computational Linguistics, Proceedings of the Conference, 2010, pp. 1317–1325.

[33] M. A. Maleq Khan, Fast distance metric based data mining techniques using p-trees: k-nearest-neighbor classification and k-clustering, 2001.

[34] E. Hovy, L. Gerber, U. Hermjakob, M. Junk, C.-Y. Lin, Question answering in webclopedia, in: Proceedings of the TREC-9 Conference, 2000.