Social Question Answering: Textual, User, and Network Features for Best Answer Prediction

PIERO MOLINO, IBM Watson, USA
LUCA MARIA AIELLO, Yahoo Labs London, UK
PASQUALE LOPS, Universitá degli Studi di Bari Aldo Moro, Italy

Community question answering (CQA) sites use a collaborative paradigm to satisfy complex information needs. Although the task of matching questions to their best answers has been tackled for more than a decade, the social question-answering practice is a complex process. The factors influencing the accuracy of question-answer matching are many and hard to disentangle. We approach the task from an application-oriented perspective, probing the space of several dimensions relevant to this problem: features, algorithms, and topics. We gather under a learning to rank framework the most extensive feature set used in literature to date, including 225 features from five different families. We test the power of such features in predicting the best answer to a question on the largest dataset from Yahoo Answers used for this task so far (40M answers) and provide a faceted analysis of the results along different topical areas and question types. We propose a novel family of distributional semantics measures that most of the time can seamlessly replace widely used linguistic similarity features, being more than one order of magnitude faster to compute and providing greater predictive power. The best feature set reaches an improvement between 11% and 26% in P@1 compared to recent well-established state-of-the-art methods.

Categories and Subject Descriptors: I.2 [**Artificial Intelligence**]: Natural Language Processing; H.3.4 [**Information Systems**]: Systems and Software—*Information networks*

General Terms: Experimentation, Performance, Algorithms

Additional Key Words and Phrases: Community question answering, best answer prediction, Yahoo Answers, distributional semantics, expert finding, expertise networks

ACM Reference Format:
Piero Molino, Luca Maria Aiello, and Pasquale Lops. 2016. Social question answering: Textual, user, and network features for best answer prediction. ACM Trans. Inf. Syst. 35, 1, Article 4 (September 2016), 40 pages.
DOI: http://dx.doi.org/10.1145/2948063

1. INTRODUCTION

Community question answering (CQA) sites such as Yahoo Answers, Stack Overflow, or Ask.com have greatly contributed to the rise of the Social Web, as they allow complex information needs to be satisfied through a collaborative paradigm. Articulated queries submitted to those systems in form of questions are processed by the crowd that generates and returns multiple possible answers. The quality of the responses

This research was partially supported by the European Community's Seventh Framework Programme FP7/2007-2013 under the SocialSensor project; by the Spanish Centre for the Development of Industrial Technology under the CENIT program, project CEN-20101037 (http://www.cenitsocialmedia.es) "Social Media"; and by grant TIN2009-14560-C03-01 of the Ministry of Science and Innovation of Spain.
Authors' addresses: P. Molino, IBM Watson, Yorktown Heights, NY, USA, 10598; L. M. Aiello, Yahoo Labs, 125 Shaftesbury Avenue, WC2H 8HR, London (UK); P. Lops, Università degli Studi di Bari "Aldo Moro", Via E. Orabona, 4, I70126 - Bari, Italy.

DOI: http://dx.doi.org/10.1145/2948063

is in turn evaluated by community members who manually match each question to its best answer. This collective process can serve high-quality answers to queries that are most often answered unsatisfactorily by conventional search engines [Morris et al. 2010]. CQA sites achieve effective results driven by the members' altruism and by their intrinsic motivation connected to the act of knowledge sharing [Nam et al. 2009; Jin et al. 2013], possibly reinforced by ad hoc incentive mechanisms [Jain et al. 2009].

The whole question-answering (QA) practice in CQA sites relies almost entirely on human actions, mainly because the process of knowledge gathering required to answer a question is hard to automate, especially for very articulated queries [Lin and Katz 2003; Andrenucci and Sneiders 2005]. However, the task of matching a question to its best answer from a corpus of human-generated responses—as well as the more general task of ranking questions by quality—has received much attention in the past decade, motivated by several practical applications.

First, automated best answer selection can make up for the shortcomings that human agents have when approaching this task. Users might not have enough time or attention span to consider all of the answers or to carefully evaluate their quality. Users may often designate as best the answers that are provided quickly and that contain sufficient level of detail, but better answers can emerge as the life cycle of a QA thread evolves, with early answers being improved and new responses being added later in time [Anderson et al. 2012]. Some questions might also receive many answers in a relatively short time, making it difficult to manually spot high-quality ones with no support of an automated ranking system. In addition, like all open collaborative platforms, CQA sites are open to abuse and user misconduct [Gyongyi et al. 2007; Dearman and Truong 2010], to which the best answer selection process is not immune.

Second, reputation systems that are implemented on most CQA platforms could benefit from mechanisms of objective quality assessment of answers. For example, in sites where the best answer is elected by voting, the occurrence of the Matthew effect (the "rich get richer" phenomenon) is hardly avoidable, leading to the emergence of a single highly voted answer, among others. However, users' answering to the same question tends to be similar in terms of both expertise and delivered quality of their answers [Anderson et al. 2012]. Therefore, rewarding mechanisms based on explicit community feedback only can unfairly disadvantage participants responding later in time.

Last, and perhaps most importantly, automatic best answer selection is a fundamental building block for social search services [Freyne et al. 2007; Evans and Chi 2008]. Social search is a very broad and multifaceted concept of which investigation is still in its early stages. One of its important principles is to leverage the expertise of the crowd and the knowledge it generates to satisfy complex information needs, also taking into account the social context. Not surprisingly, the few services that have attempted to implement this paradigm are smart variations of classic CQA systems [Horowitz and Kamvar 2010]. In this context, best answer selection is key to match questions to the most appropriate user-generated content across multiple knowledge bases and more effectively compared to standard information retrieval approaches. Additionally, machine assessment of the best answer is especially needed when the information is drawn from structured corpora where no explicit quality feedback is given by the users (e.g., topical discussion fora).

For these reasons, the tasks of best answer selection and ranking have been explored extensively over the past years (see Section 2 for an overview). Several ranking and classification algorithms trained using multiple families of features have been benchmarked against ground truths extracted from CQA portals. As new features and algorithms were explored, the performance of the results increased quite steadily. Nonetheless, the factors influencing the accuracy of question-answer matching are

plenty and entangled, often making comparison between different approaches and results difficult.

First, the feature space relevant to this task is vast and nontrivial to delve into, as it includes signals from many different domains, including information retrieval, natural language processing (NLP), and network analysis. As a result, as we shall touch on later, several research efforts have focused on selected feature subspaces in depth, whereas fewer studies have attempted to provide a more holistic view. Comparison between different results is made even more problematic when platform-specific features are considered, which might sensibly improve the performance in particular case studies but do not generalize well. On the other hand, as we shall discuss, not all feature categories that are informative for this type of task have yet been explored. Moreover, although a variety of algorithms have been appraised in each of the two main approaches to the problem—classification and ranking of best answers—systematical algorithmic comparisons still remain more infrequent in the literature. Last, whereas previous research has focused on maximizing target performance metrics (e.g., precision), the computational complexity of feature extraction has been far less discussed, thus leaving an open question about whether it is "worth the effort" to use some feature families.

In addition to the complexity of the algorithmic and feature scope, the nature of the dataset is a major element that can steer experimental outcomes. CQA communities are broadly divided into *focused*, namely specialized on a well-defined area of knowledge (e.g., computer programming), and *nonfocused* (or general purpose). Within each community type, the questions submitted can belong to a variety of subtopics. Depending on the area of knowledge and type of forum, the factors that determine a good answer can vary greatly [Agichtein et al. 2008].

Addressing all of the preceding issues is clearly challenging. In this work, we contribute to shed more light on some of these issues with a study that stretches both in breadth and depth. Given a question in input, we address the task of ranking by quality a set of available answers to maximize the likelihood that the best response for the input query is returned on top.

To approach the problem from a very practical, application-oriented perspective, we probe the space of several dimensions simultaneously—features, algorithms, and topics—as no previous work has done before. We gathered from Yahoo Answers the largest collection of question-answer pairs collected so far for this task (more than 7M questions and nearly 40M answers), and we use the vastest selection of features explored up to now, including 225 signals belonging to five distinct families: text quality, linguistic similarity, distributional semantics, user characteristics, and network structure. The distributional semantics features that we include are novel for this task, and many of features (e.g., most of the network-based ones) have never been used in combination with the others. All features that we consider abstract from the specific forum type so that the model we learned could be trained on any CQA community where there are textual questions and responses, with a best answer selected among them. Moreover, as Yahoo Answers has a general-purpose scope, we do not specialize our study on a specific type of question-answer class. Instead, we investigate the performance of feature families across four topical clusters of questions, automatically extracted from simple and general features, and across two forms of questions that have been commonly considered in the literature: general type of questions and manner questions. We rank answers with learning to rank (which has been shown to be a very effective approach for ranking in this context [Surdeanu et al. 2011]) comparing a variety of machine learning tools for training: logistic regression (LR), ListNet, RankSVM, and random forests (RF).

To summarize, we believe that ours is the first study of best answer prediction (by ranking) that is done at very large scale, on a general-purpose CQA forum, using the

largest feature space to date (including novel features), and with a study on question topics and question types and by exploration of several learning algorithms for the learning to rank framework.

Main results and findings include the following:

—Textual features are the most informative ones. However, we find that the very costly and widely used family of textual similarity features has almost no additional predictive power when our newly proposed (and much faster to compute) set of distributional semantics features is included in the model.
—Network features are somehow orthogonal to other feature types, yielding a sensible increase in performance, albeit more modestly than other signals. The most effective network features are not the ones that have been considered most extensively in previous work but are instead those based on the concept of competition network.
—The feature informativeness varies quite dramatically across question types. Text quality features are more suited to predict the best answer for factual and subjective questions, whereas features from the user profile are more predictive for discussion and poll-type questions.
—Our supervised model tops three of the latest yet already widely popular methods for best answer prediction. The most effective combination of features reaches up to 26% performance gain on precision at 1 (P@1) *over the best state-of-the-art methods*.

The article is organized as follows. After a review of the related work (Section 2), we first describe in detail the feature families that we consider (Section 3). We then describe the learning to rank framework used to combine the features together (Section 4). After introducing the Yahoo Answers dataset that we collected (Section 4.2), and the baselines (Section 4.3), we outline the experimental results in Section 4.4. We provide a discussion about the relative performance over the baselines, and a comparison between feature sets and across question types, before the final remarks in Section 5.

2. RELATED WORK

Next, we discuss some background work in the field, describing studies on CQA and expert findings, as well as some of the features that are most widely adopted in previous approaches.

2.1. Community and Nonfactoid Question Answering

Several approaches have been developed for finding and ranking answers in CQA.

One of the earliest and most widely known approaches adopts different measures of text quality to find the best answer for a given question [Agichtein et al. 2008]. Intrinsic answer properties such as grammatical, syntactic, and semantic complexity, punctuation, and typographical errors are adopted, along with question-answer similarity and user expertise estimations. We build on that work by picking all of the features reported as most effective, expanding them with new categories of features, and using a more robust learning algorithm.

A consistent branch of this research field has focused on nonfactoid QA systems, especially with regard to *Why* and *How* questions. They often use CQA datasets for evaluations and adopt similar architectures to the CQA answer-ranking engines, although focusing more on linguistic features. The importance of linguistic features for nonfactoid QA has been assessed in several studies [Verberne et al. 2008, 2010; Verberne et al. 2011], showing how the adoption of semantic role labeling–based features [Bilotti et al. 2010] and deep and shallow syntactical structures [Severyn and Moschitti 2012] can improve the performance of a nonfactoid QA system. In our experiments, we also adopt distributional linguistic features, adding even more levels of lexicalization to the linguistic representation.

Another line of approaches uses machine translation (MT) models to learn how to reformulate a question into an answer so that the probability of the translation of question into the answer can be calculated and the candidate answers can be ranked accordingly [Berger et al. 2000; Echihabi and Marcu 2003; Riezler et al. 2007]. Recently, matrix factorization algorithms have been adopted for the same goal [Zhou et al. 2013]. We adopt MT features, learning different translation models for different linguistic representations.

The study dealing with the largest-scale dataset has been done by Suredeanu et al. [2011]. They combined a large amount of features, bringing together linguistic features—those based on translation and classical frequency and density ones. They tested their ranking model on a subset of Yahoo Answers showing the effectiveness of each feature subset. As illustrated in Section 4.2, we compare our method to theirs on the same dataset (Yahoo Answers Manner Questions), adding distributional semantics, text quality, expertise network, and user-based features that were not previously considered.

In more recent years, new approaches based on lexical semantics emerged. Solutions leveraging Wikipedia entities [Zhou et al. 2013] have also been used, showing potential in addressing the retrieval of synonyms and hypernyms. Recurrent neural network language models [Yih et al. 2013] have been studied as well, confirming that lexical semantics is suitable to tackle the problem.

2.2. Expert Finding

A consistent branch of the studies on expert finding consists of casting the problem into an information retrieval problem, using methods to model the relevance of candidate users to a given question or topic. In profile-based methods, candidates are described by a textual profile and profiles ranked with respect to an expertise query [Liu et al. 2005; Craswell et al. 2001], whereas in document-based approaches, documents relevant to the query are retrieved first and then candidates are ranked based on the co-occurrence of topic and candidate mentions in the retrieved documents [Balog et al. 2006; Serdyukov and Hiemstra 2008].

Experimentation with several slight variants to such approaches has taken place over the few past years, including topic-specific information retrieval approaches, where users' expertise is calculated only from the portion of their past history that is relevant to the question [Li et al. 2011]. The use of topic modeling [Riahi et al. 2012] and classification approaches [Zhou et al. 2012] as opposed to information retrieval have been explored as well. Most often, these approaches rely on features of a single type or on quite sparse sets of features of multiple types.

In some cases, the task of expert finding has been addressed from a slightly different perspective that goes under the label of "question recommendation," which aims to recommend interesting questions for a contributor who is willing to provide answers. Such approaches tend to privilege the perspective of the answerer, such as trying to assign questions to people who have never answered, to guarantee higher fairness of the system [Kabutoya et al. 2010]. One of the most complete pieces of work in this direction uses a combination of collaborative filtering and content-based approaches, showing that the content signal is the most powerful to predict good user-question associations [Dror et al. 2011].

As an alternative to text-based methods that rely on probabilistic frameworks or topic models [Liu et al. 2010], network-based approaches can be leveraged to spot the users who are the most "expert" with respect to a specific question. Graph-based models are particularly suited to capture the expertise of individual contributors as they interact with their peers, not only limited to CQA portals.

In any social domain, the expertise may emerge from the complex interactions of users and can be modeled with the so-called expertise networks [Zhang et al. 2007a; Zhang et al. 2007b], whose construction and structure is domain dependent and can potentially mix heterogeneous graphs [Smirnova and Balog 2011; Bozzon et al. 2013]. Examples of expertise networks include scientific collaboration networks [Lappas et al. 2009], social networks [Zhang et al. 2007a; Zhang et al. 2007b; Bozzon et al. 2013], communication networks [Dom et al. 2003; Fu et al. 2007], folksonomies [Noll et al. 2009], and so on. Specifically, in CQA, as we will detail in Section 3.5, the expertise networks have been modeled based on the asker-replier information [Jurczyk and Agichtein 2007], the assignment of the best answer [Bouguessa et al. 2008; Gyongyi et al. 2007], and the competition between answerers [Liu et al. 2011; Aslay et al. 2013]. In CQA, once the experts in specific domains are identified, algorithms of question routing can be used to deliver relevant questions to them, also taking into account their availability [Li and King 2010; Horowitz and Kamvar 2010] and workload balance among the group of experts [Chang and Pal 2013].

Properties of expertise networks such as their shape, connectivity, and associativity patterns have been investigated in depth in previous work [Chen et al. 2006; Zhang et al. 2007a; Jurczyk and Agichtein 2007; Smirnova and Balog 2011]. In CQA specifically, studies on expertise networks include the analysis of user behavior in terms of topical focus and discussion triggering [Gyongyi et al. 2007], the characterization of the type of topics discussed [Adamic et al. 2008], and the relation of tie strength with the effectiveness of the given answers [Panovich et al. 2012].

However, previous literature in CQA has focused mostly on how networks of expertise could be leveraged to find the most expert users, as experts can likely provide high-quality answers. The common assumption is that graph centrality on expertise network is correlated with expertise, and this has indeed been shown extensively in the context of CQA [Jurczyk and Agichtein 2007; Aslay et al. 2013]. Standard centrality metrics, such as PageRank and HITS, as well as custom scores like ExpertiseRank [Zhang et al. 2007b], are commonly used for this purpose. Although in the past centrality metrics in CQA expertise networks have been found to be less effective in the task of best answer prediction compared to simple baselines such as the personal best answer count or ratio or best answer ratio [Chen and Nayak 2008; Bouguessa et al. 2008], recent work has shown that some combinations of expertise network and centrality metrics can indeed also beat the best answer ratio, especially for some categories of questions [Aslay et al. 2013].

In network-based frameworks, expertise can be interpreted as topic independent, similarly to the notion of authority on a graph, but expertise in CQA is more often topic dependent. To address that, a possible solution is to narrow down the focus on topic-induced subgraphs of the whole expertise network, assuming that all users who participate in it are relevant to the topic [Campbell et al. 2003; Aslay et al. 2013]. Alternatively, hybrid text network approaches can be used, either with linear combinations of scores modeling subject relevance and user expertise [Kao et al. 2010] or by adopting topic modeling to measure the relevance of the past users' reply history to a specific topic and link analysis to estimate their authority within that topic [Zhu et al. 2011]. We tackle this problem by accounting topic relevance with textual features and expertise with network features, combining them in a learning to rank fashion.

Last, we point out that although we focus on centrality-based expert finding, alternative network-oriented approaches have also been explored, such as label propagation or random walk algorithms [Fu et al. 2007; Serdyukov et al. 2008] or supervised approaches [Bian et al. 2009; Chen et al. 2012].

2.3. Comprehensive Approaches

Very few studies considered combinations of different types of features. The idea of using user interactions, network-based features, and quality estimators together for ranking the answers was introduced by Bian et al. [2008]. More recently, the same approach was reproposed, with more features and a more robust learning to rank algorithm over Stack Overflow data [Dalip et al. 2013], focusing on features specifically designed for that dataset, such as code blocks analysis. Our approach follows the path of mixing features coming from different fields and adopts the same learning to rank algorithm, but at the same time we introduce several new features, including deeper linguistic ones and those that are expertise based, dropping the ones that are too dataset specific to preserve generality, and we evaluate our approach on a larger-scale dataset.

3. FEATURES FROM CQA SITES

In this section, we describe the five main families of features that can be extracted from most of CQA sites. We will use them to train a learning to rank model aimed at the prediction of the best answer to a question. The first three families (*text quality*, *linguistic similarity*, and *distributional semantics*) belong to the macrogroup of textual features. Those features rely on the assumption that the similarity between the question and the answer and the intrinsic quality of the answer's text are good proxies for the quality of the answer itself. The last two, *user* and *expertise* network features, reflect the intrinsic quality of users in answering a question by capturing either their historical information or their interactions with other members of the community. Next we give an overview of each family; the full list of features for each group is reported in the Appendix.

3.1. Text Quality (tq)

Text quality features aim to estimate the intrinsic quality of an answer by capturing objective properties of the text composition. A summary follows.

Visual properties. This group of features quantitatively measures some properties of the text. The features belonging to this group count the number of whitespace violations (presence of multiple contiguous whitespaces or missing spaces after a punctuation mark) and the whitespace density in the text of the answer. The same counts are produced for capital letters and capitalization violations, punctuation density and violations, the URLs in the text, the parts of the answer enclosed between quotation marks, and so on. The number of capitalized words and the total count of punctuation marks are also counted, for a total of 23 features that are widely adopted in the literature [Agichtein et al. 2008; Dalip et al. 2013]. (A full feature list is provided in Table VIII).

Readability. These features evaluate how easy is to read an answer. They consider the average word length in terms of number of characters and syllables and the ratio of complex words in the answer. They also include commonly used readability indices such as Kincaid, Ari, Coleman-Liau, Flesch, Fog, Lix, and Smog, for a total of 16 features that have been already tested in previous work on CQA [Agichtein et al. 2008; Dalip et al. 2013]. The readability indices are modeled to capture the education degree or the number of years of study necessary to understand a text. In practice, they all combine heuristically quantitative metrics, such as the average length of the sentences and average length of the words, the number of characters and syllables, count of multisyllable words, and the presence of the words in whitelists. (A full feature list is provided in Table IX.)

Informativeness. This group of features was considered because a reasonable answer must contain some information that is not in the question, so we adopt three simple features that count the amount of nouns, verbs, and adjectives occurring in the answer but not in the question. (A full feature list is provided in Table X.)

3.2. Linguistic Similarity (ls)

To the best of our knowledge, the most complete approach for generation of linguistic similarity features has been considered by Surdeanu et al. [2011]. They adopt different levels of linguistic representation of a text that can be obtained using NLP algorithms to construct tokens that are then given in input to different similarity and overlap measures. This part of our work follows their approach.

The analysis of both questions and candidate answers with an NLP pipeline allows us to build representations of the text using different lexicalization levels: words, stems, lemmas, lemma and PoS tag concatenations, named entities, and super-senses as tokens. Specifically, we used the implementation offered by ClearNLP[1] v.3.2. The representations are lists of token *n*-grams. As an example, the sentence "The man plays the piano," after stopword removal, can be represented as word unigrams (*man*, *plays*, *piano*) or as lemma+pos unigrams (*man:NN*, *play:VBZ*, *piano:NN*) or as super-sense bigrams (*noun.person-verb.competition*, *verb.competition-noun.artifact*).

We also tag the text with dependency parsing and semantic role labeling [Gildea and Jurafsky 2002], so we can extract chains from them in the same way that we extract the *n*-grams. For the dependency parsing, the chains are constructed in the form of *dependent-relationType-head*, but we can also extract more general chains that do not contain the relationType. For the semantic role labeling, the chain has the form of *predicate-argumentType-argument*. Additionally in this case, the argument type can be omitted. The length of the chain can be increased, concatenating the chains of length one that share intermediate elements. For example, by concatenating unlabeled dependencies from the previous example, we obtain the chains *man-plays* and *piano-plays*.

Because longer chains do not usually add valuable information because of their sparsity [Surdeanu et al. 2011], we decided to not adopt them. The tokens that compose the chain can also be at different lexicalization degrees, but to minimize the sparsity we adopted only lemmas and super-senses. As for our example, from the sentence "The man plays the piano," we extract labeled dependencies lexicalized with lemmas (*piano-dobj-play*, *man-nsubj-play*), their unlabeled versions (*piano-play*, *man-play*), and the versions with super-sense lexicalization (*noun.artifact-dobj-verb.competition*, *noun.person-nsubj-verb.competition*) and (*noun.artifact-verb.competition*, *noun.person-verb.competition*). The same is done with the semantic role labeling annotations: the possible chains are with argument labels with lemma lexicalization (*play-A0-man*, *play-A1-piano*), without argument labels with lemma lexicalization (*play-man*, *play-piano*), with argument labels and super-sense lexicalization (*verb.competition-A0-noun.person*, *verb.competition-A1-noun.artifact*), and without argument labels with super-sense lexicalization (*verb.competition-noun.person*, *verb.competition-noun.artifact*).

To compare and assess how linguistically similar a question is to the candidate answer, we obtain the chains at different lexicalization levels for both them and then apply a similarity metric to the obtained chains.

For example, we want to compare the question "Is Guinness a kind of beer?" with the passage "Guinness produces different kinds of beers." We extract the chains of lemma bigrams (excluding stopwords) for the question and obtain *[be_guinness, guinness_kind, kind_beer]*. We do the same for the passage and obtain *[guinness_produce,*

[1]https://github.com/clir/clearnlp.

produce_different, different_kind, kind_beer]. A simple similarity metric could be the number of common tokens; in this case, we have one common tokens *kind_beer*.

Next, we list all of the similarity metrics that we apply to the chains.

Overlap. The overlap features count the ratio of tokens in common between the question and the answer as $\frac{|t_q \cap t_a|}{|t_q|}$, where t_q is the set of tokens belonging to the question and t_a is the set of tokens belonging to the answer. With this simple overlap formula, we calculate the overlap of unigrams with all of the different lexical levels, resulting in 6 features. The other 15 features are obtained calculating the overlap of 2-grams, 3-grams, and 4-grams of all lexicalizations. We consider named entities already as *n*-grams and do not split them further, as any subset of tokens would disrupt the meaningful association of words of that entity.

We also calculate the overlap of the dependency chains and semantic role labeling chains, both labeled and unlabeled and both with lemma and super-sense lexicalizations, resulting in eight features. For the different lexicalizations of the unigrams, we also calculate the Jaccard index as $\frac{|t_q \cap t_a|}{|t_q \cup t_a|}$, resulting in additional six features. We do not calculate the Jaccard index for the *n*-grams and for the dependency and semantic role labeling chains because of their sparsity. (A full feature list is provided in Table XI.)

Frequency. We use standard information retrieval techniques to obtain a measure of similarity between question and answer that takes into account the frequency of the tokens in the texts and in the whole corpus. We assign scores to the question-answer pairs according to the Tf-Idf weighting scheme, the BM25 weighting scheme, and the language modeling (with Dirichlet priors [Zhai and Lafferty 2001]) for all of the different lexicalization levels except for the named entities, for a total of 15 features. (A full feature list is provided in Table XII.)

Density. We adopt a slight modification of minimal span weighting (MSW) proposed by Monz [2004]. MSW is a proximity-based metric for document retrieval, based on a linear combination of (i) the minimal size (or span) of a text excerpt that covers all terms in common between the query and the document, (ii) the ratio of query terms that match the document, and (iii) the global text similarity between the query and the document, computed with the Lnu.ltc weighting scheme [Buckley et al. 1995].

The text similarity intercepts global similarity, the span intercepts local similarity, and the matching term ratio counterbalances the local similarity. For example, in the case in which only one query term of five matches the document, the span component would return a value of 1, whereas the matching term would be $\frac{1}{5}$. To obtain a high local similarity, the highest number of terms from the question should be present in the smallest span of terms in the answer.

As we capture the concept of global similarity with a whole set of other features (e.g., frequency based), we retain only the local similarity part, resulting in the following formula:

$$\left(\frac{\mid t_q \cap t_a \mid}{1 + \max(mms) - \min(mms)}\right)\left(\frac{\mid t_q \cap t_a \mid}{\mid t_q \mid}\right), \tag{1}$$

where t_q and t_a are the sets of tokens of the question and the answer, respectively, and $\max(mms)$ and $\min(mms)$ are the initial and final location of the shortest sequence of answer tokens containing all question tokens. We calculate it for all of the different lexicalization levels, thus obtaining six features. (A full feature list is provided in Table XIII.)

Machine translation. Research in MT, a subfield of computational linguistics, investigates the use of computational methods to translate text from one language to another.

Due to the availability of aligned corpora, statistical approaches to MT have rapidly grown in the past decade, leading to better phrase-based translations. The objective of MT in CQA is to "bridge the lexical chasm" between the question and the answer. We calculate the probability of the question being a translation of the answer $P(Q \mid A)$ and use it as a feature:

$$\begin{aligned} P(Q \mid A) &= \prod_{q \in Q} P(q \mid A) \\ P(q \mid A) &= (1-\lambda) P_{ml}(q \mid A) + \lambda P_{ml}(q \mid C) \\ P_{ml}(q \mid A) &= \sum_{a \in A} (T(q \mid a) P_{ml}(q \mid A)), \end{aligned} \tag{2}$$

where the probability that the question term q is generated from answer A, $P(q \mid A)$, is smoothed using the prior probability that the term q is generated from the entire collection of answers C, $P_{ml}(q \mid C)$, and λ is the smoothing parameter. $P_{ml}(q \mid C)$ is computed using the maximum likelihood estimator.

As the translation of a word to itself $P(w \mid w)$ is not guaranteed to be high, we set $P(w \mid w) = 0.5$ and rescale $P(w' \mid w)$ for all other w' terms in the vocabulary to sum up to 0.5 so that $\sum_{w' \in W}(w' \mid w) = 1$. This is needed for the adoption of translation models for retrieval tasks, as the exact world overlap of question and answer is a good predictor [Surdeanu et al. 2011].

Calculating the translation models for all lexicalization degrees and for all combinations of dependencies and semantic role labeling chains, we obtain 14 features. (A full feature list is provided in Table XIV.)

Others. We consider four additional miscellaneous features: the length of the exact overlap of the sequences of words in the question and the answer normalized by the length of the question, the length ratio of the question and the answer, the inverse of the length of the answer, and the inverse of the length of the question. (A full feature list is provided in Table XV.)

3.3. Distributional Semantics (ds)

In addition to features that have been used in previous work, we propose to use distributional semantics features for the first time in the context of best answer prediction.

Distributional semantics models (DSMs) have been increasingly used in computational linguistics and cognitive science. These models represent word meanings through contexts: different meanings of a word can be accounted for by looking at the different contexts in which the word occurs. Philosophical insight of distributional models can be ascribed to Wittgenstein's quote "the meaning of a word is its use in the language" [Wittgenstein 1953]. The idea behind DSMs can be summarized as follows: if two words share the same linguistic contexts, they are somehow similar in their meaning. For example, analyzing the sentences "drink a glass of wine" and "drink a glass of beer," we can assume that the words *wine* and *beer* have similar meaning because they appear in proximity to the same set of tokens (drink, a, glass, of).

This insight can be implemented with a geometrical representation of words as vectors in a semantic space. Each term is represented as a vector whose components are the words occurring in the contexts in which that term appears; the words in the vector are weighted by the number of contexts in which they occur. The granularity of the context can vary from an arbitrarily small window of neighboring terms up to the whole set of terms in the document.

For a detailed analysis of the motivations, philosophical background, and practical use of DSMs, please refer to Karlgren and Sahlgren [2001].

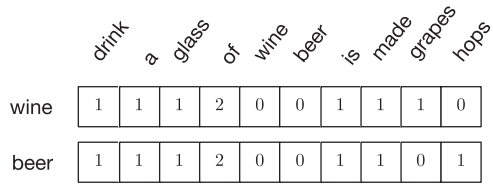

Fig. 1. Example of vector representation of words in a DSM.

As an example, given the sentences "drink a glass of wine," "wine is made of grapes," "drink a glass of beer," and "beer is made of hops," and considering words occurring in the same sentence as context, we can represent the words *wine* and *beer* with the vectors shown in Figure 1, simply counting the number of occurrences.

Semantic spaces have important advantages over other textual features. They do not require specific text operations—only tokenization is always needed. They are also language agnostic and independent of the specific corpus. This implies low computational cost and independence from any external source.

The earliest and simplest formulation of such a space has a root in the vector space model [Salton et al. 1975], which is one of the earliest models in information retrieval. Since then, they have been used in several NLP tasks [Basile 2011; Collobert et al. 2011; Turian et al. 2010], including synonym choice [Landauer and Dumais 1997], semantic priming [Landauer and Dumais 1997; Burgess et al. 1998; Jones and Mewhort 2007], finding similarity of semantic relations [Turney 2006; Turney and Littman 2005], essay grading [Wolfe et al. 1998; Foltz et al. 1999], automatic construction of thesauri [Schütze and Pedersen 1995], and word sense induction [Schütze 1998]. A useful survey of the use of vector space models for semantic processing of text has been done by [Turney and Pantel 2010]; an analysis of some compositional operators is described in the work by Mitchell and Lapata [2010]. Naturally, also several applications to information retrieval exist [Widdows and Ferraro 2008], including term-doc matrix reduction [Deerwester et al. 1990] and ambiguity resolution [Schütze and Pedersen 1995; Basile et al. 2011].

So far, DSMs have not been used in any task directly related to CQAs. Nevertheless, the ability of these models to capture paradigmatic relations between words is particularly convenient to match answers to questions, when the pure syntactic similarity could not always capture the relatedness of concepts. Next, we first describe how we build the semantic space, then we describe the DSM that we adopt, and finally we describe our strategy to integrate it inside our best answer predictor.

Co-Occurrence matrix construction. Our semantic spaces are modeled by a co-occurrence matrix. The linguistic context taken into account is a window w of co-occurring terms. In our experiments, we adopt a window of size 5 centered on the current term. Given a reference corpus[2] and its vocabulary V, an $n \times n$ co-occurrence matrix is defined as the matrix $\mathbf{M} = (m_{ij})$ whose coefficients $m_{ij} \in \mathbb{R}$ are the number of co-occurrences of the words t_i and t_j within a predetermined distance w.

The *term* $\times$ *term* matrix $\mathbf{M}$, based on simple word co-occurrences, represents the simplest semantic space, called the *term-term co-occurrence matrix* (TTM).

An example *term* $\times$ *term* matrix $\mathbf{M}$ is shown in Figure 2. The corpus from which it is obtained are the same four sentences of Figure 1: "drink a glass of wine," "wine is made of grapes," "drink a glass of beer," and "beer is made of hops."

In the literature, several methods to approximate the original matrix by rank reduction have been proposed. Dimensionality reduction allows for the discovery of

[2]The corpus could be the collection of documents indexed by the QA system but also some external text collection.

	drink	a	glass	of	wine	beer	is	made	grapes	hops
drink	0	2	2	2	1	1	0	0	0	0
a	2	0	2	2	1	1	0	0	0	0
glass	2	2	0	2	1	1	0	0	0	0
of	2	2	2	0	2	2	2	2	1	1
wine	1	1	1	2	0	0	1	1	1	0
beer	1	1	1	2	0	0	1	1	0	1
is	0	0	0	2	1	1	0	2	1	1
made	0	0	0	2	1	1	2	0	1	1
grapes	0	0	0	1	1	0	1	1	0	0
hops	0	0	0	1	0	1	1	1	0	0

Fig. 2. Example of *term* × *term* matrix $\mathbf{M}$.

high-order relations between entries and cancels noisy co-occurrences. We exploit four methods for building our reduced semantic spaces: latent semantic analysis (LSA), random indexing (RI), LSA over RI, and continuous skip-gram models. All of these methods produce a new matrix $\hat{\mathbf{M}}$, which is a $n \times k$ approximation of the co-occurrence matrix $\mathbf{M}$ with n row vectors corresponding to vocabulary terms, whereas k is the number of reduced dimensions.

Latent semantic analysis. LSA [Deerwester et al. 1990] is based on the singular value decomposition (SVD) of the original matrix $\mathbf{M}$. $\mathbf{M}$ is decomposed in the product of three matrices $\mathbf{U}\Sigma\mathbf{V}^\top$, where $\mathbf{U}$ and $\mathbf{V}$ are orthonormal matrices whose columns are the right and left eigenvectors of the matrices $\mathbf{M}^\top\mathbf{M}$ and $\mathbf{M}\mathbf{M}^\top$, respectively, whereas Σ is the diagonal matrix of the singular values of $\mathbf{M}$ placed in decreasing order.

SVD can be applied to any rectangular matrix, and if r is the *rank* of $\mathbf{M}$, then the matrix $\widetilde{\mathbf{M}} = \mathbf{U_k}\Sigma_k\mathbf{V_k}^\top$ of rank $k \ll r$, built choosing the top k singular values, is the best rank k approximation of $\mathbf{M}$. The approximated $\widetilde{\mathbf{M}}$ is shown in Figure 3.

Since the matrix $\mathbf{M}\mathbf{M}^\top$ corresponds to all possible combinations of any two terms, it is possible to compute the similarity between two terms by exploiting the relation

$$\mathbf{M}\mathbf{M}^\top = \mathbf{U}\Sigma\mathbf{V}^\top\mathbf{V}\Sigma^\top\mathbf{U}^\top = \mathbf{U}\Sigma\Sigma^\top\mathbf{U}^\top = (\mathbf{U}\Sigma)(\mathbf{U}\Sigma)^\top.$$

In the case of the k-approximation of $\mathbf{M}$, the complexity of the computation of the similarity between any two terms is reduced.

Random indexing. We exploit RI, introduced by Kanerva et al. [1988], for creating the DSM based on RI. This technique allows us to build a semantic space with no need for matrix factorization, because vectors are inferred using an incremental strategy. Moreover, it allows one to efficiently solve the problem of reducing dimensions, which is one of the key features used to uncover the latent semantic dimensions of a word distribution.

RI is based on the concept of random projection according to which randomly chosen high dimensional vectors are "nearly orthogonal." This yields a result that is comparable to orthogonalization methods, such as SVD [Landauer and Dumais 1997], but saving computational resources.

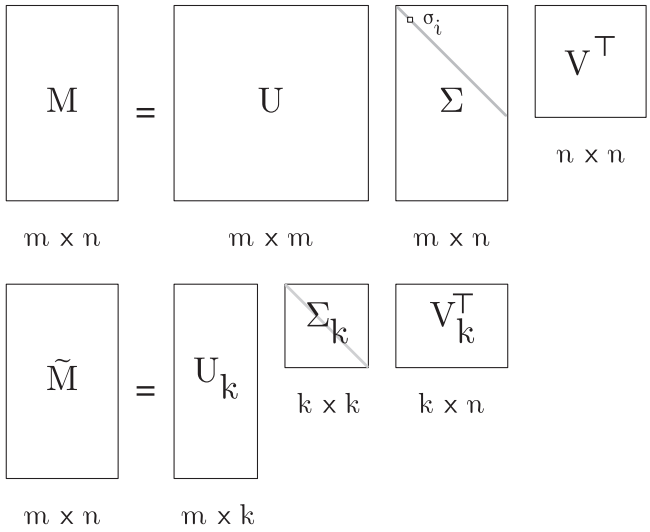

Fig. 3. Depiction of SVD matrices.

Sahlgren [2005] provide a clear and motivated introduction to RI, whereas a more detailed dissertation about RI, its construction, and the syntagmatic and paradigmatic use of context can be found in the work of Sahlgren [2006]. In the work of Cohen et al. [2010], scalability issues are discussed in detail, along with the suggestion of the capability of RI to find implicit relations among words.

Formally, given an $n \times m$ matrix $\mathbf{M}$ and an $m \times k$ matrix $\mathbf{R}$ made up of m k-dimensional random vectors, we define a new $n \times k$ matrix $\mathbf{M}'$ as follows:

$$\mathbf{M}'_{n,k} = \mathbf{M}_{n,m}\mathbf{R}_{m,k} \quad k \ll m. \tag{3}$$

The new matrix $\mathbf{M}'$ has the property to preserve the distance between points. This property is known as the Johnson-Lindenstrauss lemma [Johnson and Lindenstrauss 1984]: if the distance between any two points of $\mathbf{M}$ is d, then the distance d_r between the corresponding points in $\mathbf{M}'$ will satisfy the property that $d_r = c \cdot d$ (where c is a constant). A proof of that property has been done by Dasgupta and Gupta [1999].

The product between $\mathbf{M}$ and $\mathbf{R}$ is not actually computed, but it corresponds to building $\mathbf{M}'$ incrementally as follows:

(1) Given a corpus, a random vector is assigned to each term. The random vector is high dimensional, sparse, and with very few elements with nonzero values $\{-1, 1\}$, which ensures that the resulting vectors are nearly orthogonal, and the structure of this vector follows the hypothesis behind the concept of random projection.
(2) The semantic vector of a term is given by summing the random vectors of terms co-occurring with the target term in a predetermined context (document/sentence/window).

An example of the construction of the term vectors is shown in Figure 4.

LSA over RI. Computing LSA on the co-occurrence matrix $\mathbf{M}$ can be a computationally expensive task, as the vocabulary V can reach thousands of terms. Here we propose a simpler computation based on the application of the SVD factorization to $\mathbf{M}'$, the reduced approximation of $\mathbf{M}$ produced by RI. Sellberg and Jonsson [2008] followed a similar approach for the retrieval of similar FAQs in a QA system. Their experiments

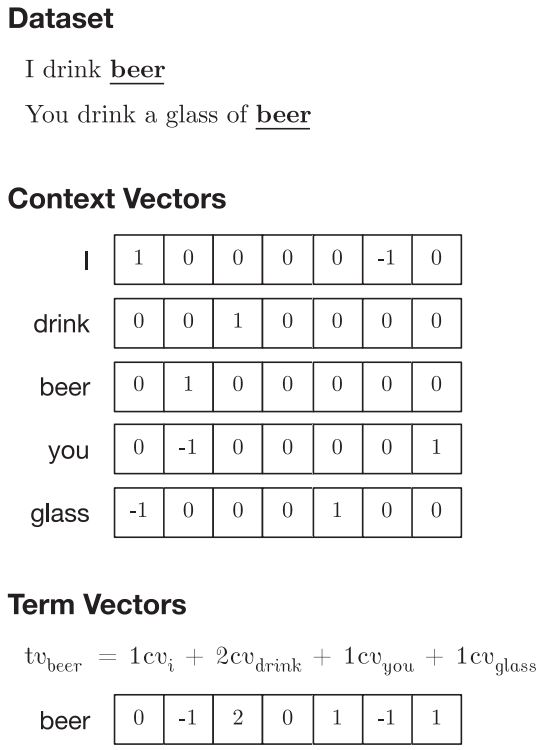

Fig. 4. Term vector construction in RI. Context vectors are random vectors.

showed that reducing the original matrix by RI resulted in a drastic reduction of LSA computation time, at the cost of a very slight decrease of performance.

Continuous skip-gram model. A quite different DSM aims at learning distributed representations of words with neural networks, because they have better performances than LSA in preserving linear regularities among words [Mikolov et al. 2013b] and the latest models are computationally less expensive, so they scale better on large datasets.

Mikolov et al. [2013a] construct a really scalable log-linear classification network, using a simpler architecture than previous work, where neural networks are usually constructed with several nonlinear hidden layers [Bengio et al. 2003]. Two such simpler networks are proposed in that work: the continuous bag-of-words model and the continuous skip-gram model. Although both are shown to be effective in semantic-syntactic word relationship learning and sentence completion tasks, the former is faster to train, whereas the latter has better performance at the cost of slightly longer training time. Even though both are really scalable, for our experiments we decided to adopt the latter one for its accuracy.

The continuous skip-gram model builds on feedforward neural networks [Bengio et al. 2003], but it consists only of input, projection, and output layers, so removing the hidden layer. As most of the complexity is caused by the nonlinear hidden layer, this improves the learning efficiency at the expenses of a representation that might be less precise but enables the learning of models with bigger amounts of data. The model, shown in Figure 5, iterates over the words in the dataset and uses each word w_t as an input to a log-linear classifier with a continuous projection layer. What it outputs is a prediction of the words within a certain range before and after the input word.

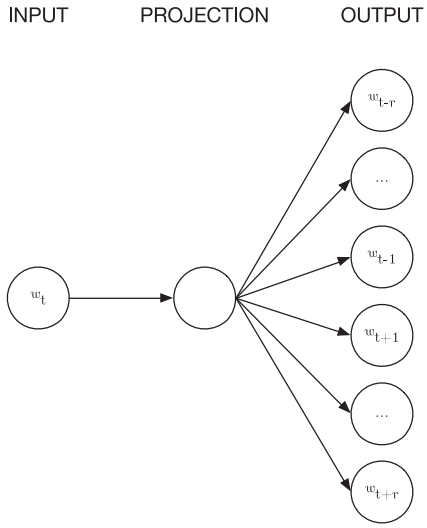

Fig. 5. Architecture of the continuous skip-gram model.

As the words that are more distant from the input word are less related to it, the model adopts a randomization policy: if c is the fixed range before and after a word, a value r is obtained picking randomly a value between $[1, c]$. Then r words before the current and r words after the current are used as correct labels, from w_{t-r} to w_{t-1} and from w_{t+1} to w_{t+r}. Randomizing the window size with a random value between 1 and c is a way to avoid overfitting: setting a fixed parameter c might indeed bias the final result, whereas having different models with a random r in $[1, c]$ smooths that risk.

At the end of the training phase, the weights associated with the projection layer are used as vector representations for each word. The resulting encoding captures meaningful word representations, where words of similar meaning have nearby representations.

DSM integration in QA. We now discuss how to leverage the word vector representations to match questions to the best answers. We use word vector representations for building the sentence-level vector representation by summing the vectors of the words that appear in the sentence. This way, we obtain vector representations for questions and answers and can compute their cosine similarity to obtain a semantic similarity measure. This measure becomes one feature used in the ranking of the answers. Questions and answers are usually short pieces of text, and this makes this strategy more suitable.

In DSMs, given the vector representation of two words $\mathbf{u} = (u_1, \ldots, u_n)^\top$ and $\mathbf{v} = (v_1, \ldots, v_n)^\top$, it is always possible to compute their similarity as the cosine of the angle between them:

$$\cos(\mathbf{u}, \mathbf{v}) = \frac{\sum_{i=1}^{n} u_i v_i}{\sqrt{\sum_{i=1}^{n} u_i^2 \sum_{i=1}^{n} v_i^2}}. \tag{4}$$

However, the user's question and the candidate answer are sentences composed by several terms. To compute the similarity between them, we need a method to compose the words occurring in these sentences. It is possible to combine words through vector addition (+). This operator is similar to the superposition defined in connectionist systems [Smolensky 1990] and corresponds to the pointwise sum of components:

$$\mathbf{s} = \mathbf{u} + \mathbf{v}, \tag{5}$$

where $s_i = u_i + v_i$.

Addition is a commutative operator, which means that it does not take into account any order or underlying structures existing between words in both questions and answers. We do not exploit more complex methods to combine word vectors, as they do not clearly outperform the simple vector addition [Mitchell and Lapata 2010]. For a deeper analysis of compositionality in distributional semnatics and its connection with syntax and formal semantics, refer to Baroni et al. [2014].

Given a phrase or sentence s, we denote with $\mathbf{s}$ its vector representation obtained applying addition operator $(+)$ to the vector representation of terms of which it is composed. Furthermore, it is possible to compute the similarity between two phrases/sentences exploiting the cosine similarity between vectors (Equation (4)).

Formally, if $q = q_1, q_2, \ldots, q_n$ and $a = a_1, a_2, \ldots, a_m$ are the question and the candidate answer, respectively, and each q_i and a_i is a term present in them, we build two vectors, $\mathbf{q}$ and $\mathbf{a}$, which respectively represent the question and the candidate answer in a semantic space. Vector representations for the question and answer are built applying the addition operator to the vector representation of words belonging to them:

$$\begin{aligned} \mathbf{q} &= q_1 + q_2 + \cdots + q_n, \\ \mathbf{a} &= a_1 + a_2 + \cdots + a_m. \end{aligned} \tag{6}$$

The similarity between $\mathbf{q}$ and $\mathbf{a}$ is computed as the cosine similarity between them.

For example, we want to compare the question $\mathbf{q}$ "Is Guinness a kind of beer?" with the passage $\mathbf{a^1}$ "Guinness produces different kinds of stouts" and the passage $\mathbf{a^2}$ "Apple produces different kinds of computers." The vector representations of the (nonstopword) words are as follows:

$v_{\text{is}} = [0.1, 0.2, 0.3, 0.25]$
$v_{\text{guinness}} = [0.7, 0.1, 0.12, 0.09]$
$v_{\text{kind}} = [0.2, 0.1, 0.65, 0.5]$
$v_{\text{beer}} = [0.8, 0.05, 0.1, 0.12]$
$v_{\text{produces}} = [0.3, 0.4, 0.1, 0.04]$
$v_{\text{different}} = [0.1, 0.21, 0.1, 0.12]$
$v_{\text{kinds}} = [0.22, 0.08, 0.67, 0.48]$
$v_{\text{stouts}} = [0.82, 0.04, 0.11, 0.11]$
$v_{\text{apple}} = [0.44, 0.71, 0.24, 0.14]$
$v_{\text{computers}} = [0.05, 0.84, 0.2, 0.6]$.

It is easy to see how the vectors for *beer* and *stout* and the vectors for *kind* and *kinds* are quite similar to each other (i.e., close in the semantic space).

The representation for $\mathbf{q}$, $\mathbf{a^1}$, and $\mathbf{a^2}$ are the following:

$$\mathbf{q} = v_{\text{is}} + v_{\text{guinness}} + v_{\text{kind}} + v_{\text{beer}} = [1.8, 0.45, 1.17, 0.96]$$
$$\mathbf{a^1} = v_{\text{guinness}} + v_{\text{produces}} + v_{\text{different}} + v_{\text{kinds}} + v_{\text{stouts}} = [2.14, 0.83, 1.1, 0.84]$$
$$\mathbf{a^2} = v_{\text{apple}} + v_{\text{produces}} + v_{\text{different}} + v_{\text{kinds}} + v_{\text{computers}} = [1.11, 2.24, 1.31, 1.38].$$

The cosine similarity among the $\mathbf{q}$ and the two passages $\mathbf{a^1}$ and $\mathbf{a^2}$ is as follows:

$\cos(\mathbf{q}, \mathbf{a^1}) = 0.9846$
$\cos(\mathbf{q}, \mathbf{a^2}) = 0.7794$.

Thus, $\mathbf{a^1}$ would be ranked higher than $\mathbf{a^2}$ in a rank list.

For computing the distributional semantics features for this set of experiments, we construct the $\mathbf{M}$ matrix using Wikipedia as a corpus and using the set of all answers

in the training set obtained from the Yahoo Answers 2011 dataset that we use for the evaluation (see Sections 4.2 and 4.4). We do so to use both general-purpose texts incorporating commonsense knowledge and knowledge that is specific to the dataset that we want to actually use. The number of dimensions of the vector representations for all methods is 400, stopwords are removed, and only unigrams are considered. We calculate the cosine similarity scores using vectors from the three types of semantic spaces constructed on both corpora, resulting in eight features. (A full set of features is provided in Table XVI.)

3.4. User Features (u)

A considerable part of the features are related to the user-centric activity to capture their behavior and history. The question and answer history and some standard fields from the public profile description are usually available in all major CQA platforms. We also assume that questions are tagged with a category, which is the case for most of the communities that enforce a strict category of systems or allow the possibility of collaborative tagging. If a categorization is not available, topic models could be used to extract it. Although most of the features that we present here have been used in prior literature of best answer selection [Agichtein et al. 2008], the decomposition of the same features across different question categories has never been explored in this context. The subgroups of user features are summarized next.

User profile. The user profile contains information that might be a good proxy for the level of user involvement in the community. These include the presence of a resume, a textual self-description of the user, a title and profile picture (surprisingly, a remarkably good estimator of expertise [Gînsca and Popescu 2013]), and the amount of time the user has been registered on the platform at the time the question was asked (we refer to it as age for simplicity), for a total of five features. (A full set of features is provided in Table XVII.)

Questions and answers. The number of questions the user asked, deleted, answered, flagged, and starred, and their normalized versions by user age, are the basic indicators for user activity. In addition to that, we also compute the ratio of those values divided by all questions asked. We replicate the same features that we calculated on the questions asked by the user on the answers given by the user as well, adding features about the thumbs up and down received by the answers and their ratio and delta. Overall, we define 19 features for the questions and 19 for the answers. (A full set of features is provided in Table XVIII.)

Question categories. We replicate the same features defined for the question and answer history of the user but only consider the category of the question actually asked. For example, if the question belongs to the category "sports," we count the questions asked and the answers given by the users in that category. This will help us estimate the user expertise and how much the user is engaged in the specific topic rather than his generic expertise or interest in different topics than the one in which the asker is interested. Thus, we add an additional 19 features for questions in the category and another 19 for the answers in the category. We also add 3 additional features that consider the entropy H of discrete probability distribution p obtained by counting the number of questions, the number of answers, and the combined number of question and answers in all of the different categories ($\|p\|$ is the number of categories).

$$H(p) = -\sum_{i=0}^{\|p\|} p_i \log_2 p_i \tag{7}$$

This allows us to evaluate how specific (high entropy) or spread out (low entropy) the user knowledge (or interest) is. (A full set of features is provided in Table XIX.)

Behavioral. Other features are related to the user behavior on the system. We count how many positive and negative votes are provided, plus their deltas and ratios, and we measure the answering speed as the temporal gap between the time of the question and answer publications, and so on, for a total of eight features. (A full set of features is provided in Table XX.)

3.5. Expertise Network Features (n)

The network features that we propose arise from expert finding literature, where a content-agnostic analysis of the interactions between participants in CQA is shown to help rank people by their general expertise in answering questions. For instance, users who provided high-quality answers (i.e., marked as best answers) to many questions will likely provide good answers in future interactions as well. Additionally, the estimation of the users' expertise may not only depend on their direct interactions but also from the interactions of other users in a recursive fashion. For example, one might imagine that, given a specific domain of knowledge, correct answering of a question made by an expert is a better indication of expertise than answering a question made by a newbie.

These considerations have motivated past research in the study of expertise networks [Zhang et al. 2007a], especially for CQA. Expertise networks are weighted graphs where nodes are users and weighted edges model interactions that account for the flow of activity, knowledge, or status differences among peers. In the past, three main expertise networks were defined and studied for CQA. We provide visual examples for each in Figure 6.

The first is the asker-replier network [Jurczyk and Agichtein 2007], where directed edges flow from askers to answerers and are weighted by the number of replies. The second is the asker–best answerer network (ABAN) [Bouguessa et al. 2008; Gyongyi et al. 2007], where directed edges flow from askers to the best answerers and are weighted by the number of best answers given. The last is the competition-based expertise network (CBEN) [Aslay et al. 2013], where edges flow between all users who answered the same question toward the user who gave the best answer to that question; the possibility of building such a network is conditioned by the possibility for the users to explicitly mark the best answer, which is most often true in large-scale CQAs. The advantage of ARN is that it needs less information to be built, but ignoring the signal coming from the best answer, it considers all answers to have equal value. ABAN addresses this problem; however, on the other hand, it disregards the information of people who answered and whose answer was not selected as the best. CBEN was proposed to take into account both aspects and to capture at the same time the inherent competition that exists between answerers to get awarded with the best answer. In addition, no relation between asker and answerer is represented in CBEN under the assumption that asking a question is not necessarily related to a lack of expertise [Zhang et al. 2007a; Zhang et al. 2007b], especially in broad general-purpose QA communities.

The application of graph centrality metrics to the expertise networks mentioned earlier produces a ranking of the users based on their expertise. Depending on the specific combination of network and centrality, the ranking might convey different meanings; however, in all cases, users with higher scores are supposed to have higher expertise compared to their peers with lower scores. In previous work, this assumption was validated by multiple experiments and some specific network-centrality combinations (PageRank on ARN, indegree on ABAN, HITS on CBEN) have proved to work best in the task of best answer prediction [Aslay et al. 2013]. In this work, in a learning to rank

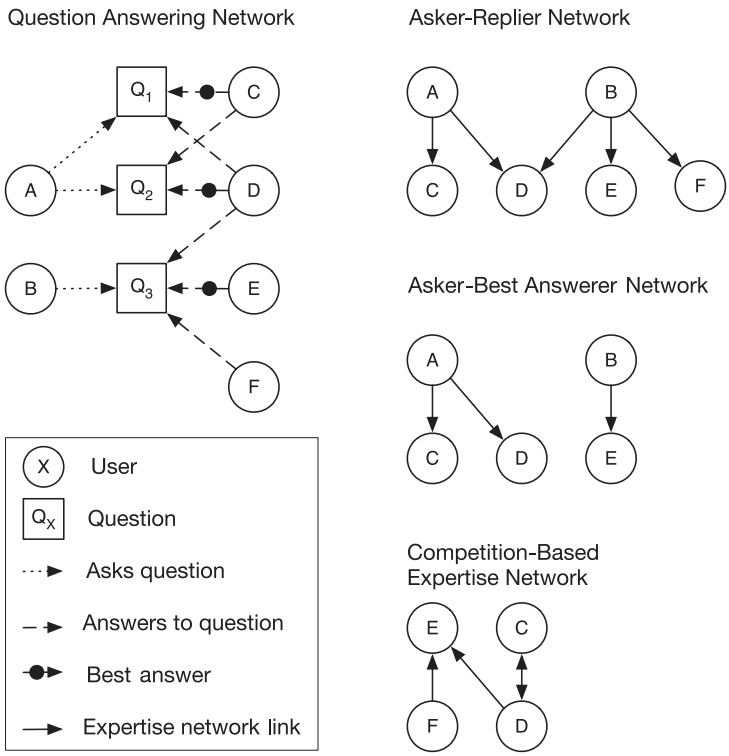

Fig. 6. Graph of relations between askers, questions, and answerers (left) and the three types of expertise networks derived by it (right).

framework, we aim to include a wide set of features, and therefore we do not restrict ourselves to specific pairs but consider instead all combinations of expertise networks (ARN, ABAN, CBEN) with the centrality metrics that have been applied to them in past work (PageRank [Page et al. 1999], HITS [Kleinberg 1999], indegree) for a total of nine features. We consider networks built on the full question-answer dataset with no distinction of topic, as we want to measure general expertise with network features and account for relevance with the textual features. (A full set of features is provided in Table XXI.)

4. EXPERIMENTAL EVALUATION

Next we describe the problem under study and the framework we use to address it, along with four baselines against which we compare our method.

Problem statement. Given in input a question q and the set of its answers $A(q)$, among which exactly one answer $a^* \in A(q)$ has been selected as the best answer, output a rank of the answers in the set $A(q)$ that has a high likelihood of a^* being placed high in the rank. This problem is a generalization of the best answer selection and can be reduced to it if only the first element in the ranking is considered, but it allows a more detailed analysis of the results and a richer comparison between methods.

4.1. Learning to Rank for Best Answer Prediction

We address the problem using a learning to rank approach, where question-document pairs (q, d) are labeled with relevance judgments that indicate the degree of relevance of the document d with respect to query q. Each pair is represented by a set of features

that are usually an indication of the degree of similarity between q and d, yet also information about q and d in isolation, such as their length or the PageRank of Web documents. Each pair is treated as a single data point, and a set of data points can be used for training purposes to learn a function to predict the best ranking of different documents according to a query.

Several algorithms have been proposed for this goal in the literature [Liu 2011]. We opted for RF [Breiman 2001] because of its resilience to overfitting, a problem that may affect our experimental setting due to the size of our dataset, and because of the successful results in several use cases related to CQA [Dalip et al. 2013] and in other large-scale retrieval experiments [Mohan et al. 2011].

Let $x_i = \phi(d, q)$, where ϕ is a feature extractor and x_i is an m-dimensional vector. Let $D = (x_1, y_1), \ldots, (x_n, y_n)$ be a set of query-document pairs x_i and their associated relevance ratings $y_i \in Y$. In the specific case of our question-answer dataset, the relevance is maximum for the best answer and zero for all other answers.

The RF algorithm trains a model H such that $H(x_i) \approx y_i$ and so that the ranking of all documents d appearing in pair with a query q according to $H(x_i)$ is similar to the ranking according to y_i. The pseudocode of the procedure is listed in Algorithm 1.

ALGORITHM 1: Random Forests

Require: $D = (x_1, y_1), \ldots, (x_n, y_n), r > 0$
1: **for** $i \leftarrow 1$ to r **do**
2: $D_t \leftarrow sample(D)$
3: $K \leftarrow roandomPick(m)$
4: $h_i \leftarrow buildDecisionTree(D_t, K)$
5: **end for**
6: $H() \leftarrow \frac{1}{r} \sum_{i=1}^{r} h_i()$
7: **return** $H()$

The main idea of RF is to apply a prediction tree—specifically, a regression tree in our case—to M subsets of D and then average the results. A sample D_t is extracted with replacement from D (step 2). A set K of features is randomly picked from the feature set so that $|K| \leq m$ (step 3). A regression tree is induced from D_t using the features in K (step 4). The whole process is repeated r times, and the outputs of all single trees are averaged to obtain the function H (step 6). The use of different samples of the data from the same distribution and of different sets of features for learning the individual regression trees prevents the overfitting.

In our experiments, the queries are the questions and the documents are the candidate answers. In our evaluation, we use the implementation provided by the RankLib library.[3]

4.2. Dataset

The instance of CQA that we consider for our experiments is Yahoo Answers because of its popularity and richness of content. Launched in 2005, it is one of the largest general-purpose CQA services to date, hosting questions and answers on a broad range of topics, categorized through a predefined two-level taxonomy. There are 26 predefined top-level categories (TLCs), such as politics, sports, or entertainment, and a growing number of leaf-level categories (LLCs)—more than 1,300 at the time of this study—such as makeup or personal finance. Similarly to other CQA portals, Yahoo Answers follows a strict question-answer format, with questions submitted as short statements

[3]http://sourceforge.net/p/lemur/wiki/RankLib/.

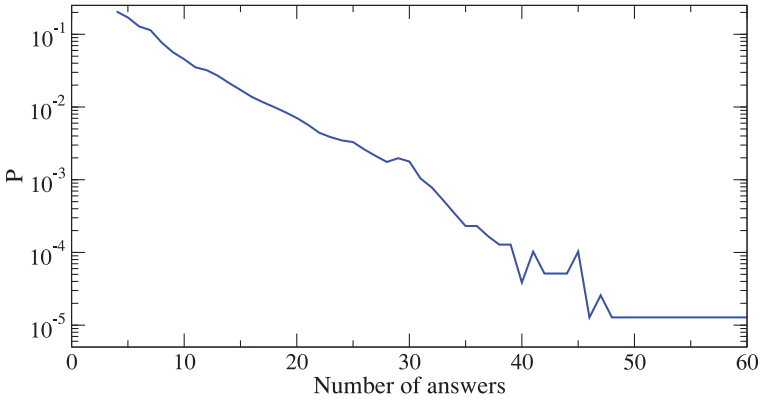

Fig. 7. Distribution of number of answers per question. Questions with fewer than four answers are not included in the dataset.

with optional detailed description and a mandatory LLC that is assigned by the asker. Questions have a life cycle of states that goes from *open* to *voting* and finally to *resolved*, and users can actively moderate content using several feedback mechanisms, such as by marking spam or abusive content, adding stars to interesting questions, voting for best answers, and giving thumbs-up or thumbs-down ratings to answers. Among all feedback signals, the most important is the selection of the best answer, which is designated by the asker, or if the asker does not provide it after a given time, it is selected by the community by majority vote. The process of best answer selection is important not only to reward contributors according to the Yahoo Answers incentive scheme[4], but also for archival purposes, as the best answer will be given evidence in the page and will serve users who might have the same question in the future.

4.2.1. Yahoo Answers 2011. We first collected a data sample from Yahoo Answers related to the period between January and December 2011, for a total of >7.2M resolved questions with the best answer assigned by the asker, >39.5M answers, and >6.1M unique users. As our goal is to select the best answers among the ones provided, we need to consider only questions with a minimum number of answers for the task to be meaningful. For this reason, all answers that we selected for the dataset have at least 4 answers. The distribution of number of answers per question is shown in Figure 7. The dataset contains the text of the question and answers, their metadata (timestamp, question category, number of thumbs up and down, best answer mark) and the metadata associated to the user involved in the process (user self-description, subscription date, number of questions asked and answers given, number of best answers, presence of thumbnail photo in the profile). Each question has only one answer marked as the best one.

As Yahoo Answers is a general-purpose portal, not only does it cover different topics, but it also hosts a broad variety of question types. In practice, every forum category has some mix of requests for factual information, advice seeking, and social conversation or discussion [Harper et al. 2009]. The most refined categorization obtained on Yahoo Answers so far has been proposed by Aslay et al. [2013], who extended the seminal work by Adamic et al. [2008] and used k-means to cluster Yahoo Answers LLCs using features such as the average number of replies to a question and the average

[4]A new user is granted 100 points, and asking a question costs 5 points. Several user actions are worth new points, among which the submission of an answer that is the most rewarding one (as it is worth 10 points). Detailed scheme are available at http://answers.yahoo.com/info/scoring_system.

number of characters in a reply, and some activity-based features such as the proportion of questions with contradictory answer ratings (thumbs up vs. thumbs down). The optimal R^2 was obtained for $k = 4$, corresponding to the following main question types: factual-information seeking (31% of the questions), subjective-information seeking (32%), social discussion (10%), and poll-survey conducting (27%). We use this categorization to compare the feature performance also across question types.

4.2.2. Yahoo Answers Manner Questions. To compare our results directly against some state-of-the-art methods, we decided to replicate the experiments with a publicly available dataset[5] that contains a sample of manner questions collected from the U.S. Yahoo Answers site. Manner questions are those questions that ask how to do something. Following what was done in previous work [Surdeanu et al. 2011], the manner questions are extracted following two simple heuristics that aim at preserving only high-quality questions and answers. This is done by retaining all questions that (i) match the regular expression, `how (to | do | did | does | can | would | could | should)`, and (ii) have at least four words, out of which at least one is a noun and at least one is a verb. This process yields 142,627 questions and 771,938 answers, with an average of 5.41 answers for each question.

4.3. Baselines

We compare our approach with four different baselines:

—*BM25*: Standard ranking function used in information retrieval to rank matching documents according to their relevance to a given search query. We consider the question as query and the answers as documents. We chose this baseline over other information retrieval baselines because it is the best-performing one in our dataset.
—*Finding high-quality content in social media* [Agichtein et al. 2008]: A supervised method trained on measures of text quality, such as grammatical, syntactic, and semantic complexity; punctuation and typo errors, and simple question-answer similarity and user expertise estimations. Readability and informativeness are also included. Their best performance was achieved using stochastic gradient boosted trees. We replicated their learning approach and feature set. This baseline was selected because it was the state of the art for best answer selection on Yahoo Answers data.
—*Exploiting user feedback to learn to rank answers* [Dalip et al. 2013]: The learning to rank approach for ranking answers in Q&A fora using RF, trained on several families of features. We train it using 142 features overall, excluding those that in the original publication were specifically designed for the Stack Overflow use-case and all features related to HTML formatting of the question and answer, as we do not have text format information in our dataset.
—*Learning to rank answers* [Surdeanu et al. 2011]: Combines linguistic features, those based on translation, classical frequency, density ones, and Web-correlation–based ones with a learning to rank approach, carried out with an averaged perceptron. It was applied on the Yahoo Answers Manner Questions dataset as a testbed. The authors did not use any user-based feature nor expertise-based ones, as this kind of information is missing from the dataset, but they also did not adopt text-quality features that we adopt, and the levels of lexicalizations of their linguistic features are only terms, lemmas, and super-senses. We chose this baseline because it was the state of the art on the Yahoo Answers Manner Questions dataset for P@1.
—*Improved answer ranking* [Hieber and Riezler 2011]: Similar to the previous one, this work relies mainly on textual features, but adopting Piggybacking features on Web snippets. The ranking is done adopting an SVM-based ranker. Their evaluation was

[5]http://webscope.sandbox.yahoo.com/catalog.php?datatype=l.

carried out on the Yahoo Answers Manner Questions dataset as well. We chose this baseline because it was the state of the art on the Yahoo Answers Manner Questions dataset for mean reciprocal rank (MRR).

4.4. Performance Analysis

We evaluate our learning to rank framework by performing a 10-fold cross validation. Questions in each dataset are split into a training set *Tr*, a test set *Ts*, and a validation set *Vs*. Applying 10-fold cross validation means that the dataset is divided into 10 disjoint partitions. The experiment was performed in 10 steps, and at each step, eight partitions were used as training set, one partition was used as test set, and the last partition was used as validation set, which is adopted for tuning the RF hyperparameters (e.g., number of bags, number of trees, number of leaves). The steps were repeated until each of the 10 disjoint partitions was used as the *Ts*, and results were averaged over 10 runs. It is worthnoting that the validation set is used to optimize the parameters of the learning to rank algorithms, including the baseline in Dalip et al. [2013].

For each question, all of its answers are ranked by the learning to rank method. To allow a direct comparison of the quality of the ranking with results in previous work, we use three standard information retrieval metrics that have been commonly used to evaluate this task, namely mean reciprocal rank (MRR), P@1, and discounted cumulative gain (DCG). When considering the answers to a single question, these are formally defined as follows:

$$\mathrm{RR} = \frac{1}{\mathrm{rank(BA)}} \quad \mathrm{DCG_k} = \sum_{i=1}^{k} \frac{2^{rel_i} - 1}{\log_2(i+1)} \quad \mathrm{P@1} = rel_1,$$

where A is the set of answers, $rank(BA)$ is the rank of the best answer for that question, and rel_i is an indicator function of relevance that returns 1 if the answer in the i^{th} position in the ranking is the best answer. All scores are then averaged over all questions ($\frac{1}{|Q|}\sum_{q\in Q} score(q)$). In case the best answer is ranked first, $MRR = DCG = P@1 = 1$. As each question has only one answer marked as correct (the best answer), the $DCG = nDCG$, because the ideal DCG is equal to 1. Given the large size of our experimental dataset, all differences that we obtain are statistically significant under the nonparametric randomization test [Smucker et al. 2007], with $p < .01$.

4.4.1. Performance on Yahoo Answers 2011. To gain insights about the predictive power of different feature families, we train the model on several subsets of features, with a greedy selection procedure. We first separately test each family and pick the best-performing one; at the next step, we keep that family and combine it with all others to select the best combination. The process is repeated until all feature families are included. The greedy strategy allows us to find a locally optimal choice at each stage, with the hope of finding a global optimum in a reasonable time. Results are shown in Table I.

The most predictive features are the ones belonging to the **tq** family. This group includes 44 features that capture many facets of the text structure that are indeed good proxies for the answer quality. On the other hand, **n** features alone are the worst performing; this is expected, as centrality metrics capture general expertise in a content-agnostic way, so they do not embed information about the topic or structure of the questions and answers. A similar consideration can be done for the user features even though their performance is sensibly higher than the network features. This supports the findings in previous work [Chen and Nayak 2008], which found simple user features such as the percentage of best answers very predictive of the level of

Table I. Predictive Power of the Learning to Rank Framework Trained on Different Feature Subsets on the Yahoo Answers 2011 Dataset

Features	P@1	MRR	DCG
BM25	$0.4161_{\pm 3 \cdot 10^{-4}}$	$0.5549_{\pm 3 \cdot 10^{-4}}$	$0.6585_{\pm 3 \cdot 10^{-4}}$
[Agichtein et al. 2008]	$0.5256_{\pm 3 \cdot 10^{-4}}$	$0.6389_{\pm 2 \cdot 10^{-4}}$	$0.6975_{\pm 3 \cdot 10^{-4}}$
[Dalip et al. 2013]	$0.5971_{\pm 3 \cdot 10^{-4}}$	$0.7262_{\pm 2 \cdot 10^{-4}}$	$0.7931_{\pm 3 \cdot 10^{-4}}$
tq	$0.5454_{\pm 3 \cdot 10^{-4}}$	$0.7178_{\pm 3 \cdot 10^{-4}}$	$0.7815_{\pm 3 \cdot 10^{-4}}$
ls	$0.5297_{\pm 3 \cdot 10^{-4}}$	$0.7079_{\pm 3 \cdot 10^{-4}}$	$0.7768_{\pm 3 \cdot 10^{-4}}$
ds	$0.4944_{\pm 3 \cdot 10^{-4}}$	$0.6919_{\pm 3 \cdot 10^{-4}}$	$0.7722_{\pm 3 \cdot 10^{-4}}$
u	$0.5376_{\pm 3 \cdot 10^{-4}}$	$0.7165_{\pm 3 \cdot 10^{-4}}$	$0.7915_{\pm 3 \cdot 10^{-4}}$
n	$0.4582_{\pm 3 \cdot 10^{-4}}$	$0.6808_{\pm 3 \cdot 10^{-4}}$	$0.7646_{\pm 4 \cdot 10^{-4}}$
tq+u	$0.6361_{\pm 3 \cdot 10^{-4}}$	$0.7758_{\pm 3 \cdot 10^{-4}}$	$0.8416_{\pm 3 \cdot 10^{-4}}$
tq+n	$0.6021_{\pm 3 \cdot 10^{-4}}$	$0.7529_{\pm 3 \cdot 10^{-4}}$	$0.8237_{\pm 3 \cdot 10^{-4}}$
tq+ds	$0.5697_{\pm 3 \cdot 10^{-4}}$	$0.7310_{\pm 3 \cdot 10^{-4}}$	$0.8073_{\pm 3 \cdot 10^{-4}}$
tq+ls	$0.5670_{\pm 3 \cdot 10^{-4}}$	$0.7286_{\pm 3 \cdot 10^{-4}}$	$0.8056_{\pm 3 \cdot 10^{-4}}$
tq+u+n	$0.6575_{\pm 3 \cdot 10^{-4}}$	$0.7900_{\pm 3 \cdot 10^{-4}}$	$0.8533_{\pm 4 \cdot 10^{-4}}$
tq+u+ds	$0.6370_{\pm 3 \cdot 10^{-4}}$	$0.7770_{\pm 3 \cdot 10^{-4}}$	$0.8438_{\pm 3 \cdot 10^{-4}}$
tq+u+ls	$0.6357_{\pm 3 \cdot 10^{-4}}$	$0.7753_{\pm 3 \cdot 10^{-4}}$	$0.8417_{\pm 3 \cdot 10^{-4}}$
tq+u+n+ds	$0.6612_{\pm 3 \cdot 10^{-4}}$	$0.7918_{\pm 3 \cdot 10^{-4}}$	$0.8545_{\pm 3 \cdot 10^{-4}}$
tq+u+n+ls	$0.6577_{\pm 3 \cdot 10^{-4}}$	$0.7900_{\pm 3 \cdot 10^{-4}}$	$0.8528_{\pm 3 \cdot 10^{-4}}$
all	$0.6632_{\pm 3 \cdot 10^{-4}}$	$0.7954_{\pm 3 \cdot 10^{-4}}$	$0.8554_{\pm 3 \cdot 10^{-4}}$

Note: Feature families are text quality (tq), linguistic similarity (ls), distributional semantics (ds), user (u), and expertise network (n). Best feature combinations in each section of the table are in bold. The 99% confidence intervals are reported.

user expertise. Finally, **ls** features outperform the **ds** features, when used in isolation; this may be mainly due to the very different dimensionality of the feature sets, as distributional semantics include a set of just six features. Regarding the baselines, we note, as expected, that an approach that is not specifically tailored on the task like BM25 performs poorly. The method from Agichtein et al. [2008] also has a performance that is lower than the ones obtained by the single-feature families, partially because of the different training procedure but mainly because it is trained with a set of features that is smaller than the ones that we consider inside each family. The best-performing baseline is Dalip's learning to rank framework [Dalip et al. 2013], which achieve a higher precision at 1 even compared with our framework when trained on single-feature families; its superiority no longer holds when two or more feature families are combined.

When combining features in pairs, interesting patterns emerge. Even though **tq** and **ls** are the best performing individually, their combination improves the performance only slightly, as the signal that they bring is very overlapping. Indeed, their combination is the worst performing among all feature pairings. The same happens with **ds** features. Additionally, **n** and especially **u** features are instead more orthogonal to the **tq** information and are able to boost performance considerably. Most importantly, we find that **n** and **u** features carry predictive information that is nonoverlapping, as the combination of both with **tq** features results in further noticeable improvement.

Combinations of three feature groups or more make clear that despite the high informativeness on their own, the **ls** features give a fairly small contribution to the performance, and replacing them with **ds** features leads even to a small improvement. Given that the time of computation of the **ls** features is roughly 12 times more than the **ds** ones (as empirically measured in our test), it appears that **ds** features are stronger and more lightweight (they are very few) and therefore are a more viable alternative.

Table II. Ablation Test

Feature	Δ
tq: Preposition count	0.0484
tq: Verbs not in question	0.0468
tq: Nouns not in question	0.0463
tq: Unique words in answer	0.0441
tq: Pronouns count	0.0415
tq: Punctuation count	0.0406
tq: Average words per sentence	0.0402
ds: Random indexing on Yahoo Answers	0.0394
ls: Super-senses overlap	0.0371
tq: Adjectives not in question	0.0362
tq: Conjunctions count	0.0357
tq: "To be" count	0.0354
tq: Capitalized words count	0.0351
ls: Lemma overlap	0.0346
ls: Stem overlap	0.0342
tq: Auxiliary verbs count	0.0341
ls: Term overlap	0.0325
ls: Super-senses BM25	0.0318
n: Indegree on CBEN	0.0307
u: Answerer's best answer ratio	0.0304

Note: Δ measures the loss of performance in MRR when the feature is removed, when the full set of features is employed. Prefixes in names indicate the family of the feature.

The MRR score obtained with the combination of all feature groups is a 10% improvement over the baseline, whereas the P@1 score is an 11% improvement and the DCG score is an 8% improvement.

Besides the greedy aggregation of feature families, to discover which single features give the best signal for the prediction, we run an ablation test to measure the performance decrease Δ in the prediction when single features are removed from the set. The 20 ones with the highest values of Δ are reported in Table II. We note that although **tq** features tend to dominate, one feature from **ds** and one from **n** make it into the top 20 (8th and 19th, respectively).

As final remark, we note that when plotting the MRR and DCG for rankings that include the top n results only (Figure 8), we see that the values tend to increase considerably in the first positions of the ranking, meaning that the best answer, if not ranked as first, is usually ranked among the top two or three answers.

4.4.2. Performance on Yahoo Answers Manner Questions. The last two baselines that we consider (Surdeanu et al. [2011] and Hieber and Riezler [2011]) have been applied to the smaller Yahoo Answers Manner Questions dataset described in Section 4.2. To get a fair comparison with them, we replicate their same experimental setup on the same dataset and repeat the greedy feature family combination as described earlier. An RF model is learned for each feature set, and performances are reported in Table III. We performed a 10-fold cross validation exactly like the previous dataset (see Section 4.4).

All three groups improve over the baseline significantly both in P@1 and MRR, with **tq** being the most effective. It is worth noticing that the distributed representation–based feature alone can compete with the other two groups of features, which are composed of 42 features for **tq** and 74 for **ls**.

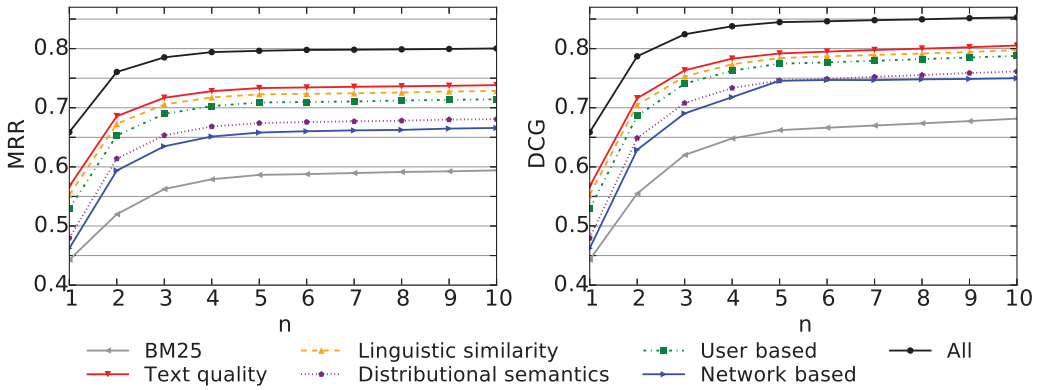

Fig. 8. MRR and DCG computed for the first n positions of the ranking, for the different features families, plus the BM25 baseline and the full set of features. The 99% confidence intervals are all in the range $[2.6 \cdot 10^{-4}, 3.6 \cdot 10^{-4}]$ and thus are too small to be represented in the figure.

Table III. Predictive Power of the Learning to Rank Framework Trained on Different Feature Subsets on the Yahoo Answers Manner Questions Dataset

Features	P@1	MRR	DCG
BM25	$0.4118_{\pm 2 \cdot 10^{-4}}$	$0.5612_{\pm 3 \cdot 10^{-4}}$	$0.6126_{\pm 3 \cdot 10^{-4}}$
Surdeanu et al. [2011]	$0.5091_{\pm 3 \cdot 10^{-4}}$	$0.6465_{\pm 3 \cdot 10^{-4}}$	-
Hieber and Riezler [2011]	$0.4844_{\pm 3 \cdot 10^{-4}}$	$0.6676_{\pm 3 \cdot 10^{-4}}$	-
ds	$0.6269_{\pm 2 \cdot 10^{-4}}$	$0.7838_{\pm 3 \cdot 10^{-4}}$	$0.8348_{\pm 3 \cdot 10^{-4}}$
ls	$0.6329_{\pm 3 \cdot 10^{-4}}$	$0.7869_{\pm 3 \cdot 10^{-4}}$	$0.8389_{\pm 3 \cdot 10^{-4}}$
tq	$0.6392_{\pm 3 \cdot 10^{-4}}$	$0.8001_{\pm 3 \cdot 10^{-4}}$	$0.8501_{\pm 3 \cdot 10^{-4}}$
ds+ls	$0.6333_{\pm 3 \cdot 10^{-4}}$	$0.7787_{\pm 3 \cdot 10^{-4}}$	$0.8384_{\pm 3 \cdot 10^{-4}}$
ds+tq	$0.6685_{\pm 3 \cdot 10^{-4}}$	$0.8071_{\pm 3 \cdot 10^{-4}}$	$0.8573_{\pm 3 \cdot 10^{-4}}$
ls+tq	$0.6552_{\pm 3 \cdot 10^{-4}}$	$0.8005_{\pm 3 \cdot 10^{-4}}$	$0.8504_{\pm 3 \cdot 10^{-4}}$
ds+ls+tq	$0.6680_{\pm 3 \cdot 10^{-4}}$	$0.8070_{\pm 3 \cdot 10^{-4}}$	$0.8576_{\pm 3 \cdot 10^{-4}}$

Note: The 99% confidence intervals are reported.

Taking into account the combinations of features, we observe that the best-performing one is the composition of **ds** and **tq**. The combination of **ds** and **ls** leads to an improvement of 0.0004 for P@1 over the **ls** group alone, a nonstatistically significant improvement. This is expected, as both groups try to intercept the topical similarity between question and answer.

The most interesting result that can be observed is that adding the **ls** group to the previous best-scoring group **ds+tq** only improves the performance of 0.0003 for **DCG**, again a nonstatistically significant improvement. This finding suggests that in this setting, the linguistic features, requiring a really expensive preprocessing time to be computed, can be substituted with a single feature based on distributed representations of words without any loss of accuracy.

Finally, the best P@1 scores obtained with the **ds+tq** and **ds+tq+ls** feature groups are a 31% improvement over the state of the art (best of the three baselines), whereas the best MRR scores obtained with the **ds+tq+ls** features group are an improvement of 19% over the state of the art.

4.4.3. Feature Analysis. We analyzed in more the detail the results of the ablation test, focusing on the newly proposed features.

Table IV. Distributional Semantics–Based Features Ablation Ranking

Feature	Rank
ds: Random indexing on Yahoo Answers	8
ds: Continuous skip-gram model on Yahoo Answers	30
ds: LSA on Wikipedia	38
ds: LSA after random indexing on Wikipedia	39
ds: Random indexing on Wikipedia	40
ds: Continuous skip-gram model on Wikipedia	41
ds: LSA after random indexing on Yahoo Answers	88
ds: LSA on Yahoo Answers	90

Table V. Network Features Ablation Ranking

Feature	Rank
n: Indegree on CBEN	19
n: Hits on CBEN	34
n: Indegree on ABAN	100
n: Hits on ABAN	108
n: Indegree on ARN	162
n: Hits on ARN	163
n: PageRank on ARN	171
n: PageRank on CBEN	183
n: PageRank on ABAN	184

Considering the features based on distributional semantics (**ds**), reported in Table IV, we can clearly see that the best-performing feature, RI on Yahoo Answers, ranks 8th. This is encouraging and suggests that the adoption of textual data coming from the dataset itself is helpful. The continuous skip-gram model on the same datasets is the second best one, ranking 30th, supporting the suggestion of the RI feature. The other two features using models learned on the same dataset rank 88th (LSA over RI) and 90th (LSA), almost in the middle of the ranking. The difference with respect to RI suggests that probably the number of dimensions (400) is not an appropriate choice for the LSA, and an optimization of this parameter could lead to improvements.

The features that adopt Wikipedia as a text source for learning the models rank really close: 38th for LSA, 39th for LSA over RI, 40th for the continuous skip-gram model, and 41st for RI. This suggests that the differences in models, in this case, are less influent than the dataset itself. As Wikipedia contains more than 4M articles, the huge quantity of text in this dataset leads to similarly behaving models.

Considering the network-based features (**n**), reported in Table V, the best-performing network structure is the CBENs. Two features based on models calculated on this network are the top ranked: indegree on CBEN is 19th, and Hits on CBEN is 34th. The same two models calculated on the ABAN are ranked in the middle of the ranking, 100th and 108th, respectively, whereas those calculated on the ARN are ranked lower in the ranking, 162nd and 163rd. The fact that both models, the simple indegree and the Hits authority, are found really close in the ranking suggests that they behave in a very similar way. At the bottom of the ranking, we found the PageRank model calculated on ARN (171st), on CBEN (183rd), and ABAN (184th). This suggests that PageRank is not a good fit in this setting and leads to quite bad results.

4.4.4. Question Categories. Different types of questions may imply different notions of a "high-quality" answer. To investigate this aspect, we get back to the bigger Yahoo Answers 2011 dataset and break down the performance of the different feature families

Table VI. MRR Scores Obtained with Single-Feature Families on the Yahoo Answers 2011 Dataset

	Factual	Subjective	Discussion	Poll
tq	**0.7490**$\pm 3 \cdot 10^{-4}$	**0.7397**$\pm 3 \cdot 10^{-4}$	0.6836$\pm 3 \cdot 10^{-4}$	0.6924$\pm 3 \cdot 10^{-4}$
ls	0.7399$\pm 3 \cdot 10^{-4}$	0.7273$\pm 3 \cdot 10^{-4}$	0.6636$\pm 3 \cdot 10^{-4}$	0.6505$\pm 3 \cdot 10^{-4}$
ds	0.7040$\pm 3 \cdot 10^{-4}$	0.6899$\pm 3 \cdot 10^{-4}$	0.6532$\pm 3 \cdot 10^{-4}$	0.6658$\pm 3 \cdot 10^{-4}$
u	0.7378$\pm 3 \cdot 10^{-4}$	0.7276$\pm 3 \cdot 10^{-4}$	**0.6880**$\pm 3 \cdot 10^{-4}$	**0.7034**$\pm 3 \cdot 10^{-4}$
n	0.7164$\pm 2 \cdot 10^{-4}$	0.7109$\pm 2 \cdot 10^{-4}$	0.6289$\pm 4 \cdot 10^{-4}$	0.6372$\pm 2 \cdot 10^{-4}$
all	0.8216$\pm 3 \cdot 10^{-4}$	0.8059$\pm 3 \cdot 10^{-4}$	0.7666$\pm 3 \cdot 10^{-4}$	0.7800$\pm 3 \cdot 10^{-4}$

Note: The 99% confidence intervals are reported.

by the four question categories that we defined in Section 4.2. For brevity, we report the values for MRR only (P@1 and DCG follow the same trends) and limit the analysis to feature families taken in isolation.

In agreement with previous work [Aslay et al. 2013], the best answer is more difficult to predict for discussion and poll-type questions, as they are naturally less suited to expert ranking. Best answers for factual and subjective questions are better surfaced by the **tq** features, whereas the **u** features are dominating discussions and polls.

Focusing on the novel features that we introduce, we note their complementary behavior, with **ds** better than **n** in polls and discussion (and even better than **ls** for polls) but worse in factual and subjective questions. In addition, it is worth noting that **ds** has the smaller variance in performance across categories. Detailed results are provided in Table VI.

4.4.5. Different Algorithms. Our decision to use a pointwise approach like RF as a ranking algorithm is based on the intuition that pairwise and listwise approaches are not likely to be more effective because of the presence of only one correct answer for each question in the dataset. This means that we have a number of equally wrong answers that we cannot distinguish based on their relevance to the answer, so the full list of answers is not likely to bring more information than the single answers. RF is supposed to be quite resilient to overfitting when applied on large-scale training sets.

To assess that RF is indeed the best approach, we run the evaluations on the same datasets with the same features using different algorithms.

We chose LR as an alternative pointwise approach because it was successfully adopted in large-scale real-world QA scenarios [Ferrucci 2011]. For pairwise approaches, we chose RankSVM [Joachims 2002] as the algorithm to test against, as SVMs were shown to be effective on the same Yahoo Answers Manner Questions dataset [Surdeanu et al. 2011]. Finally, for a listwise approach, we chose to test against ListNet [Cao et al. 2007]. For all of the algorithms, we tuned the hyperparameters from the adopted libraries (RankLib[6] and SVMLight[7])—for example, regularization and kernel for RankSVM and learning rate and number of epochs for ListNet.

The results in Table VII show only the trends for MRR using all of the features, but the same trends are also present by changing the adopted feature set combination and metric. LR is the worst-performing algorithm on all sets of questions, whereas among RankSVM and ListNet, the difference is very small, with RankSVM obtaining slightly higher results on all question sets but *Poll*. None of the alternative algorithms can

[6]http://sourceforge.net/p/lemur/wiki/RankLib/.
[7]http://svmlight.joachims.org.

Table VII. MRR Scores Obtained with Different Learning to Rank Algorithms on the Yahoo Answers 2011 Dataset

	LR	RankSVM	ListNet	RF
Manner	$0.6964_{\pm 2 \cdot 10^{-4}}$	$0.7880_{\pm 3 \cdot 10^{-4}}$	$0.7705_{\pm 3 \cdot 10^{-4}}$	$\mathbf{0.7927}_{\pm 3 \cdot 10^{-4}}$
Factual	$0.7418_{\pm 3 \cdot 10^{-4}}$	$0.7967_{\pm 3 \cdot 10^{-4}}$	$0.7814_{\pm 3 \cdot 10^{-4}}$	$\mathbf{0.8216}_{\pm 3 \cdot 10^{-4}}$
Subjective	$0.7193_{\pm 3 \cdot 10^{-4}}$	$0.7834_{\pm 3 \cdot 10^{-4}}$	$0.7600_{\pm 3 \cdot 10^{-4}}$	$\mathbf{0.8059}_{\pm 4 \cdot 10^{-4}}$
Discussion	$0.6891_{\pm 3 \cdot 10^{-4}}$	$0.7448_{\pm 3 \cdot 10^{-4}}$	$0.7245_{\pm 4 \cdot 10^{-4}}$	$\mathbf{0.7667}_{\pm 3 \cdot 10^{-4}}$
Poll	$0.7037_{\pm 3 \cdot 10^{-4}}$	$0.7479_{\pm 3 \cdot 10^{-4}}$	$0.7498_{\pm 3 \cdot 10^{-4}}$	$\mathbf{0.7800}_{\pm 3 \cdot 10^{-4}}$
All	$0.7177_{\pm 3 \cdot 10^{-4}}$	$0.7684_{\pm 2 \cdot 10^{-4}}$	$0.7650_{\pm 3 \cdot 10^{-4}}$	$\mathbf{0.7954}_{\pm 3 \cdot 10^{-4}}$

Note: The 99% confidence intervals are reported.

reach the performance levels reached by RF in any of the question sets, and this gives some empirical evidence that our choice was reasonable.

5. CONCLUSIONS

We contribute to bring order to the vast literature on the task of best answer selection by gathering the largest set of features considered for this task so far, grouped in five families, combining them with a learning to rank approach, and testing them on large datasets from Yahoo Answers. We propose a new suite of distributional semantics–based features, in combination with the textual signal and the information from several expertise networks. In addition to being able to outperform the prediction ability of state-of-the-art methods up to 26% in P@1, our experiments allow us also to draw important conclusions about the impact of different features employed that have never been spelled out in previous literature due to a lack of extensive and systematic feature comparison. We summarize our findings as follows:

—Textual features are by far the ones with higher predictive potential compared to user-centric features or to the expertise network centrality scores. This is mainly because the content of the questions and answers (their topic and structure) are a more important source of information to determine the QA match rather than the expertise of the answerers. Those features are preferred when dealing with factual-type questions.
—Among the textual features, text quality and distributional semantics are generally preferred to linguistic similarity. We indeed found that linguistic similarity's signal is mostly captured by other features already. This is an important finding, as linguistic similarity features have been used in several previous approaches but are roughly 12 times more computationally expensive than distributional semantics ones.
—The new distributional semantics–based approach that we propose achieves surprisingly good results considering the very small cardinality of its feature set.
—User and network features determine a considerable improvement over the textual-based features, and their contribution is not completely overlapping, meaning that considering network interaction rather than the individual user activity adds real value to the prediction. When user or network information is available, it is advisable to use it in combination with text-quality features instead of using different textual features combined.

We believe that our work will help to take stock of the research on the task of best answer prediction and set the basis for new developments in the field.

APPENDIX

Table VIII. Visual Property Features

Group tq	**Subgroup** visual property
Count of auxiliary verbs	
Count of pronouns	
Count of conjunctions	
Count of prepositions	
Count of occurrences of the verb “to be”	
Count of punctuation marks	
Minimum length of quoted text	
Average length of quoted text	
Maximum length of quoted text	
Number of quotes	
Number of sentences	
Number of capitalized words	
Number of characters	
Number of whitespace violations (lack or redundancy)	
Number of URLs	
Number of words	
Number of capitalization violations (i.e., no capital letter after sentence mark)	
Number of question marks	
Number of punctuation violations (lack or redundancy)	
Number of whitespaces	
Punctuation characters divided by all characters	
Whitespace characters divided by all characters	
Capital letters characters divided by all characters	

Table IX. Readability Features

Group tq	**Subgroup** readability
Average words per sentence	
Average words length in syllables	
Average words length in characters	
Number of complex words divided by all words	
Number of unique words	
Average unique words per sentence	
Flesch-Kinkaid grade level	
Automated readability index	
Coleman-Liau index	
Flesch reading ease	
Gunning-Fog index	
LIX score	
SMOG grade	
Number of short sentences	
Number of long sentences	
Automated readability index of the question	

Table X. Informativeness Features

Group tq	**Subgroup** informativeness
Number of nouns present in the answer but not in the question	
Number of verbs present in the answer but not in the question	
Number of adjectives present in the answer but not in the question	

Table XI. Overlap Features

Group ls	**Subgroup** overlap
Overlap of lemmas	
Overlap of concatenations of lemmas and PoS tags	
Overlap of named entities	
Overlap of stems	
Overlap of super-senses	
Overlap of terms	
Overlap of labeled dependencies with lemma lexicalization	
Overlap of labeled dependencies with super-sense lexicalization	
Overlap of unlabeled dependencies with lemma lexicalization	
Overlap of unlabeled dependencies with super-sense lexicalization	
Overlap of labeled semantic roles with lemma lexicalization	
Overlap of labeled semantic roles with super-sense lexicalization	
Overlap of unlabeled semantic roles with lemma lexicalization	
Overlap of unlabeled semantic roles with super-sense lexicalization	
Jaccard index of lemmas	
Jaccard index of concatenations of lemmas and PoS tags	
Jaccard index of named entities	
Jaccard index of stems	
Jaccard index of super-senses	
Jaccard index of terms	
Overlap of lemma bigrams	
Overlap of bigrams of concatenations of lemmas and PoS tags	
Overlap of stem bigrams	
Overlap of super-sense bigrams	
Overlap of term bigrams	
Overlap of lemma trigrams	
Overlap of trigrams of concatenations of lemmas and PoS tags	
Overlap of stem trigrams	
Overlap of super-sense trigrams	
Overlap of term trigrams	
Overlap of lemma tetragrams	
Overlap of tetragrams of concatenations of lemmas and PoS tags	
Overlap of stem tetragrams	
Overlap of super-sense tetragrams	
Overlap of term tetragrams	

Table XII. Frequency Features

Group ls	**Subgroup** frequency
BM25 with lemmas	
BM25 with concatenations of lemmas and PoS tags	
BM25 with stems	
BM25 with super-senses	
BM25 with terms	
Language modeling with lemmas	
Language modeling with concatenations of lemmas and PoS tags	
Language modeling with stems	
Language modeling with super-senses	
Language modeling with terms	
TF-IDF with lemmas	
TF-IDF with concatenations of lemmas and PoS tags	
TF-IDF with stems	
TF-IDF with super-senses	
TF-IDF with terms	

Table XIII. Density Features

Group ls	**Subgroup** density
Density of lemmas	
Density of concatenations of lemmas and PoS tags	
Density of named entities	
Density of stems	
Density of super-senses	
Density of terms	

Table XIV. Machine Translation Features

Group ls	**Subgroup** machine translation
Machine translation of lemmas	
Machine translation of concatenations of lemmas and PoS tags	
Machine translation of named entities	
Machine translation of stems	
Machine translation of super-senses	
Machine translation of terms	
Machine translation of labeled dependencies with lemma lexicalization	
Machine translation of labeled dependencies with super-sense lexicalization	
Machine translation of unlabeled dependencies with lemma lexicalization	
Machine translation of unlabeled dependencies with super-sense lexicalization	
Machine translation of labeled semantic roles with lemma lexicalization	
Machine translation of labeled semantic roles with super-sense lexicalization	
Machine translation of unlabeled semantic roles with lemma lexicalization	
Machine translation of unlabeled semantic roles with super-sense lexicalization	

Table XV. Other Features

Group ls	**Subgroup** other
Number of consecutive overlapping words	
Length of the answer divided by the length of question (in characters)	
1 divided by the length of the answer	
1 divided by the length of the question	

Table XVI. Distributional Semantics–Based Features

Group ls	**Subgroup** distributional semantics
Semantic similarity using the LSA on Wikipedia corpus	
Semantic similarity using the random indexing on Wikipedia corpus	
Semantic similarity using the LSA after random indexing on Wikipedia corpus	
Semantic similarity using the continuous skip-gram model on Wikipedia corpus	
Semantic similarity using the LSA on Yahoo Answers corpus	
Semantic similarity using the random indexing on Yahoo Answers corpus	
Semantic similarity using the LSA after random indexing on Yahoo Answers corpus	
Semantic similarity using the continous skip-gram model on Yahoo Answers corpus	

Table XVII. User Profile Features

Group u	**Subgroup** profile
Presence of a resume in the user profile (1 if present, 0 otherwise)	
Length of the resume (in characters)	
Presence of a title in the user profile (1 if present, 0 otherwise)	
Presence of a picture in the user profile (1 if present, 0 otherwise)	
Time since the account creation	

Table XVIII. Question-Answer Features

Group u	**Subgroup** question answer
Number of (not deleted) questions asked by the user	
Number of deleted questions asked by the user	
Number of answered questions asked by the user	
Number of flagged questions asked by the user	
Number of questions with a star asked by the user	
Number of (not deleted) questions asked by the user divided by the time since the account creation	
Number of deleted questions asked by the user divided by the time since the account creation	
Number of answered questions asked by the user divided by the time since the account creation	
Number of flagged questions asked by the user divided by the time since the account creation	
Number of questions with a star asked by the user divided by the time since the account creation	
Number of (not deleted) questions divided by all questions asked by the user	
Number of deleted questions divided by all questions of the user	
Number of answered questions divided by all questions asked by the user	
Number of flagged questions divided by all questions asked by the user	
Number of questions with a star divided by all questions asked by the user	
Minimum automatic readability index of questions asked by the user	
Maximum automatic readability index of questions asked by the user	
Average automatic readability index of questions asked by the user	
Number of questions divided by number of answers given by the user	
Number of (not deleted) answers given by the user	
Number of deleted answers given by the user	
Number of best answers given by the user	
Number of flagged questions asked by the user	
Number of (not deleted) answers given by the user divided by the time since the account creation	
Number of deleted answers given by the user divided by the time since the account creation	
Number of best answers given by the user divided by the time since the account creation	
Number of flagged answers given by the user divided by the time since the account creation	
Number of (not deleted) answers divided by all answers given by the user	
Number of deleted answers divided by all answers given by the user	
Number of best answers divided by all answers given by the user	
Number of flagged answers divided by all answers given by the user	
Number of positive votes that the answers given by the user have received	
Number of negative votes that the answers given by the user have received	
Difference of positive and negative votes that the answers given by the user have received	
Number of positive votes divided by number of negative votes that the answers given by the user have received	
Minimum automatic readability index of answers given by the user	
Maximum automatic readability index of answers given by the user	
Average automatic readability index of answers given by the user	

Table XIX. Category Features

Group u	**Subgroup** category
Number of (not deleted) questions asked by the user in the category of the question	
Number of deleted questions asked by the user in the category of the question	
Number of answered questions asked by the user in the category of the question	
Number of flagged questions asked by the user in the category of the question	
Number of questions with a star asked by the user in the category of the question	
Number of (not deleted) questions asked by the user divided by the time since the account creation in the category of the question	
Number of deleted questions asked by the user divided by the time since the account creation in the category of the question	
Number of answered questions asked by the user divided by the time since the account creation in the category of the question	
Number of flagged questions asked by the user divided by the time since the account creation in the category of the question	
Number of questions with a star asked by the user divided by the time since the account creation in the category of the question	
Number of (not deleted) questions divided by all questions asked by the user in the category of the question	
Number of deleted questions divided by all questions of the user in the category of the question	
Number of answered questions divided by all questions asked by the user in the category of the question	
Number of flagged questions divided by all questions asked by the user in the category of the question	
Number of questions with a star divided by all questions asked by the user in the category of the question	
Minimum automatic readability index of questions asked by the user in the category of the question	
Maximum automatic readability index of questions asked by the user in the category of the question	
Average automatic readability index of questions asked by the user in the category of the question	
Number of questions divided by number of answers given by the user in the category of the question	
Number of (not deleted) answers given by the user in the category of the question	
Number of deleted answers given by the user in the category of the question	
Number of best answers given by the user in the category of the question	
Number of flagged questions asked by the user in the category of the question	
Number of (not deleted) answers given by the user divided by the time since the account creation in the category of the question	
Number of deleted answers given by the user divided by the time since the account creation in the category of the question	
Number of best answers given by the user divided by the time since the account creation in the category of the question	
Number of flagged answers given by the user divided by the time since the account creation in the category of the question	
Number of (not deleted) answers divided by all answers given by the user in the category of the question	
Number of deleted answers divided by all answers given by the user in the category of the question	
Number of best answers divided by all answers given by the user in the category of the question	
Number of flagged answers divided by all answers given by the user in the category of the question	
Number of positive votes that the answers given by the user have received in the category of the question	
Number of negative votes that the answers given by the user have received in the category of the question	
Difference of positive and negative votes that the answers given by the user have received in the category of the question	
Positive/negative vote ratio for the answers given by the user have received in the category of the question	
Minimum automatic readability index of answers given by the user in the category of the question	
Maximum automatic readability index of answers given by the user in the category of the question	
Average automatic readability index of answers given by the user in the category of the question	
Entropy of the vector constructed by counting the number of questions in each category	
Entropy of the vector constructed by counting the number of answers in each category	

Table XX. Behavioral Features

Group u	**Subgroup** behavioral
Internal Yahoo Answer authority score of the user	
Number of flags given by the user	
Number of positive votes given by the user	
Number of negative votes given by the user	
Difference between the number of positive votes and the number of negative votes given by the user	
Number of positive votes divided by the number of negative votes given by the user	
Time between the question is posted and the answer is given by the user	
Number of answers given to this question	

Table XXI. Network Features

Group n	**Subgroup** arn - aban - cben
Indegree of the user in the ARN	
PageRank of the user in the ARN	
Hits Authority of the user in the ARN	
Indegree of the user in the best answerer network	
PageRank of the user in the best answerer network	
Hits Authority of the user in the best answerer network	
Indegree of the user in the CBEN	
PageRank of the user in the CBEN	
Hits Authority of the user in the CBEN	

ACKNOWLEDGMENTS

We are grateful to Çiğdem Aslay for her precious suggestions and support in gathering the data.

REFERENCES

Lada A. Adamic, Jun Zhang, Eytan Bakshy, and Mark S. Ackerman. 2008. Knowledge sharing and Yahoo Answers: Everyone knows something. In *Proceedings of the 17th International Conference on World Wide Web (WWW'08)*. 665–674.

Eugene Agichtein, Carlos Castillo, Debora Donato, Aristides Gionis, and Gilad Mishne. 2008. Finding high-quality content in social media. In *Proceedings of the International Conference on Web Search and Data Mining (WSDM'08)*. ACM, New York, NY, 183–194.

Ashton Anderson, Daniel Huttenlocher, Jon Kleinberg, and Jure Leskovec. 2012. Discovering value from community activity on focused question answering sites: A case study of stack overflow. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'12)*. ACM, New York, NY, 850–858. DOI:http://dx.doi.org/10.1145/2339530.2339665

Andrea Andrenucci and Eriks Sneiders. 2005. Automated question answering: Review of the main approaches. In *Proceedings of the International Conference on Information Technology and Applications*. 514–519. DOI:http://dx.doi.org/10.1109/ICITA.2005.78

Çigdem Aslay, Neil O'Hare, Luca Maria Aiello, and Alejandro Jaimes. 2013. Competition-based networks for expert finding. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'13)*. 1033–1036.

Krisztian Balog, Leif Azzopardi, and Maarten de Rijke. 2006. Formal models for expert finding in enterprise corpora. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'06)*. ACM, New York, NY, 43–50.

Marco Baroni, Raffaela Bernardi, and Roberto Zamparelli. 2014. Frege in space: A program of compositional distributional semantics. *Linguistic Issues in Language Technologies* 9, 6, 5–110.

Pierpaolo Basile. 2011. Super-sense tagging using support vector machines and distributional features. In *Evaluation of Natural Language and Speech Tools for Italian*. Lecture Notes in Computer Science, Vol. 7689. Springer, 176–185.

Pierpaolo Basile, Annalina Caputo, and Giovanni Semeraro. 2011. Integrating sense discrimination in a semantic information retrieval system. In *Information Retrieval and Mining in Distributed Environments*. Studies in Computational Intelligence, Vol. 324. Springer, 249–265.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. 2003. A neural probabilistic language model. *Journal of Machine Learning Research* 3, 1137–1155.

Adam L. Berger, Rich Caruana, David Cohn, Dayne Freitag, and Vibhu O. Mittal. 2000. Bridging the lexical chasm: Statistical approaches to answer-finding. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'00)*. 192–199.

Jiang Bian, Yandong Liu, Eugene Agichtein, and Hongyuan Zha. 2008. Finding the right facts in the crowd: Factoid question answering over social media. In *Proceedings of the 17th International Conference on World Wide Web (WWW'08)*. 467–476.

Jiang Bian, Yandong Liu, Ding Zhou, Eugene Agichtein, and Hongyuan Zha. 2009. Learning to recognize reliable users and content in social media with coupled mutual reinforcement. In *Proceedings of the 18th International Conference on World Wide Web (WWW'09)*. ACM, New York, NY, 51–60.

Matthew W. Bilotti, Jonathan L. Elsas, Jaime G. Carbonell, and Eric Nyberg. 2010. Rank learning for factoid question answering with linguistic and semantic constraints. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*. 459–468.

Mohamed Bouguessa, Benoît Dumoulin, and Shengrui Wang. 2008. Identifying authoritative actors in question-answering forums: The case of Yahoo! Answers. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, New York, NY, 866–874.

Alessandro Bozzon, Marco Brambilla, Stefano Ceri, Matteo Silvestri, and Giuliano Vesci. 2013. Choosing the right crowd: Expert finding in social networks. In *Proceedings of the Joint 2013 EDBT/ICDT Conferences (EDBT'13)*. ACM, New York, NY, 637–648.

Leo Breiman. 2001. Random forests. *Machine Learning* 45, 1, 5–32.

C. Buckley, A. Singhal, and M. Mitra. 1995. New retrieval approaches using SMART: TREC 4. In *Proceedings of the 4th Text Retrieval Conference (TREC-4)*. 25–48.

Curt Burgess, Kay Livesay, and Kevin Lund. 1998. Explorations in context space: Words, sentences, discourse. *Discourse Processes* 25, 2–3, 211–257.

Christopher S. Campbell, Paul P. Maglio, Alex Cozzi, and Byron Dom. 2003. Expertise identification using email communications. In *Proceedings of the 2003 ACM CIKM International Conference on Information and Knowledge Management*. ACM, New York, NY, 528–531.

Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. 2007. Learning to rank: From pairwise approach to listwise approach. In *Proceedings of the 24th International Conference on Machine Learning (ICML'07)*. ACM, New York, NY, 129–136.

Shuo Chang and Aditya Pal. 2013. Routing questions for collaborative answering in community question answering. In *Proceedings of Advances in Social Networks Analysis and Mining (ASONAM'13)*. ACM, New York, NY, 494–501.

Bee-Chung Chen, Anirban Dasgupta, Xuanhui Wang, and Jie Yang. 2012. Vote calibration in community question-answering systems. In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'12)*. 781–790.

Haiqiang Chen, Huawei Shen, Jin Xiong, Songbo Tan, and Xueqi Cheng. 2006. Social network structure behind the mailing lists: ICT-IIIS at TREC 2006 expert finding track. In *Proceedings of the 15th Test Retrieval Conference (TREC'06)*.

Lin Chen and Richi Nayak. 2008. Expertise analysis in a question answer portal for author ranking. In *Proceedings of the 2008 IEEE/WIC/ACM International Conference on Web Intelligence (IW'08)*. 134–140.

Trevor Cohen, Roger Schvaneveldt, and Dominic Widdows. 2010. Reflective random indexing and indirect inference: A scalable method for discovery of implicit connections. *Journal of Biomedical Informatics* 43, 2, 240–256. DOI:http://dx.doi.org/10.1016/j.jbi.2009.09.003

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel P. Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research* 12, 2493–2537.

Nick Craswell, David Hawking, Anne-Marie Vercoustre, and Wilkins Peter. 2001. Panoptic expert: Searching for experts not just for documents. In *Ausweb Poster Proceedings*.

Daniel Hasan Dalip, Marcos André Gonçalves, Marco Cristo, and Pável Calado. 2013. Exploiting user feedback to learn to rank answers in Q&A forums: A case study with stack overflow. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'13)*. 543–552.

S. Dasgupta and A. Gupta. 1999. *An Elementary Proof of the Johnson-Lindenstrauss Lemma*. Technical Report TR-99-006. International Computer Science Institute, Berkeley, CA.

David Dearman and Khai N. Truong. 2010. Why users of Yahoo! Answers do not answer questions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'10)*. ACM, New York, NY, 329–332. DOI: http://dx.doi.org/10.1145/1753326.1753376

Scott C. Deerwester, Susan T. Dumais, Thomas K. Landauer, George W. Furnas, and Richard A. Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American Society for Information Science* 41, 6, 391–407.

Byron Dom, Iris Eiron, Alex Cozzi, and Yi Zhang. 2003. Graph-based ranking algorithms for e-mail expertise analysis. In *Proceedings of the 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery (DMKD'03)*. ACM, New York, NY, 42–48.

Gideon Dror, Yehuda Koren, Yoelle Maarek, and Idan Szpektor. 2011. I want to answer; who has a question? Yahoo! Answers recommender system. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (DMKD'11)*. ACM, New York, NY, 1109–1117. DOI: http://dx.doi.org/10.1145/2020408.2020582

Abdessamad Echihabi and Daniel Marcu. 2003. A noisy-channel approach to question answering. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*. 16–23.

Brynn M. Evans and Ed H. Chi. 2008. Towards a model of understanding social search. In *Proceedings of the 2008 ACM Conference on Computer Supported Cooperative Work (CSCW'08)*. ACM, New York, NY, 485–494. DOI: http://dx.doi.org/10.1145/1460563.1460641

David A. Ferrucci. 2011. IBM's Watson/DeepQA. *SIGARCH Computer Architecture News* 39, 33, Article No. 1.

Peter W. Foltz, Darrell Laham, and Thomas K. Landauer. 1999. The intelligent essay assessor: Applications to educational technology. *Interactive Multimedia Electronic Journal of Computer-Enhanced Learning* 1, 2, 939–944.

Jill Freyne, Rosta Farzan, Peter Brusilovsky, Barry Smyth, and Maurice Coyle. 2007. Collecting community wisdom: Integrating social search and social navigation. In *Proceedings of the 12th International Conference on Intelligent User Interfaces (IUI'07)*. ACM, New York, NY, 52–61. DOI: http://dx.doi.org/10.1145/1216295.1216312

Yupeng Fu, Rongjing Xiang, Yiqun Liu, Min Zhang, and Shaoping Ma. 2007. Finding experts using social network analysis. In *Proceedings of the 2007 IEEE / WIC / ACM International Conference on Web Intelligence (WI'07)*. IEEE, Los Alamitos, CA, 77–80.

Daniel Gildea and Daniel Jurafsky. 2002. Automatic labeling of semantic roles. *Computational Linguistics* 28, 3, 245–288.

Alexandru-Lucian Gînsca and Adrian Popescu. 2013. User profiling for answer quality assessment in Q&A communities. In *Proceedings of the 2013 Workshop on Data-Driven User Behavioral Modelling and Mining from Social Media (DUBMODCIKM'13)*. ACM, New York, NY, 25–28.

Zoltan Gyongyi, Georgia Koutrika, Jan Pedersen, and Hector Garcia-Molina. 2007. *Questioning Yahoo! Answers*. Technical Report 2007-35. Stanford InfoLab, Stanford, CA.

F. Maxwell Harper, Daniel Moy, and Joseph A. Konstan. 2009. Facts or friends? Distinguishing informational and conversational questions in social Q&A sites. In *Proceedings of the 27th International Conference on Human Factors in Computing Systems (CHI'09)*. ACM, New York, NY, 759–768.

Felix Hieber and Stefan Riezler. 2011. Improved answer ranking in social question-answering portals. In *Proceedings of the 3rd International CIKM Workshop on Search and Mining User-Generated Contents (SMUC'11)*. ACM, New York, NY, 19–26.

Damon Horowitz and Sepandar D. Kamvar. 2010. The anatomy of a large-scale social search engine. In *Proceedings of the 19th International Conference on World Wide Web (WWW'10). Raleigh, North Carolina*. ACM, New York, NY, 431–440.

Shaili Jain, Yiling Chen, and David C. Parkes. 2009. Designing incentives for online question and answer forums. In *Proceedings of the 10th ACM Conference on Electronic Commerce (EC'09)*. ACM, New York, NY, 129–138. DOI: http://dx.doi.org/10.1145/1566374.1566393

Xiao-Ling Jin, Zhongyun Zhou, Matthew K. O. Lee, and Christy M. K. Cheung. 2013. Why users keep answering questions in online question answering communities: A theoretical and empirical investigation. *International Journal of Information Management* 33, 1, 93–104. DOI: http://dx.doi.org/10.1016/j.ijinfomgt.2012.07.007

Thorsten Joachims. 2002. Optimizing search engines using clickthrough data. In *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, New York, NY, 133–142.

William B. Johnson and Joram Lindenstrauss. 1984. Extensions of Lipschitz mappings into a Hilbert space. *Contemporary Mathematics* 26, 189–206.

Michael N. Jones and Douglas J. K. Mewhort. 2007. Representing word meaning and order information in a composite holographic lexicon. *Psychological Review* 114, 1, 1–37.

Pawel Jurczyk and Eugene Agichtein. 2007. Discovering authorities in question answer communities by using link analysis. In *Proceedings of the 16th ACM Conference on Information and Knowledge Management (CIKM'07)*. ACM, New York, NY, 919–922.

Yutaka Kabutoya, Tomoharu Iwata, Hisako Shiohara, and Ko Fujimura. 2010. Effective question recommendation based on multiple features for question answering communities. In *Proceedings of the 4th International AAAI Conference on Weblogs and Social Media*.

Pentti Kanerva. 1988. *Sparse Distributed Memory*. MIT Press, Cambridge MA.

Wei-Chen Kao, Duen-Ren Liu, and Shiu-Wen Wang. 2010. Expert finding in question-answering Web sites: A novel hybrid approach. In *Proceedings of the 2010 ACM Symposium on Applied Computing (SAC'10)*. ACM, New York, NY, 867–871.

Jussi Karlgren and Magnus Sahlgren. 2001. From words to understanding. In *Foundations of Real-World Intelligence*, Y. Uesaka, P. Kanerva, and H. Asoh (Eds.). CSLI Publications, Stanford, CA, 294–308.

Jon M. Kleinberg. 1999. Hubs, authorities, and communities. *ACM Computing Surveys* 31, 4, 5.

Thomas K. Landauer and Susan T. Dumais. 1997. A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological Review* 104, 2, 211–240.

Theodoros Lappas, Kun Liu, and Evimaria Terzi. 2009. Finding a team of experts in social networks. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, New York, NY, 467–476.

Baichuan Li and Irwin King. 2010. Routing questions to appropriate answerers in community question answering services. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management (CIKM'10)*. 1585–1588.

Baichuan Li, Irwin King, and Michael R. Lyu. 2011. Question routing in community question answering: Putting category in its place. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management (CIKM'11)*. 2041–2044. DOI: http://dx.doi.org/10.1145/2063576.2063885

Jimmy Lin and Boris Katz. 2003. Question answering from the Web using knowledge annotation and knowledge mining techniques. In *Proceedings of the 12th International Conference on Information and Knowledge Management (CIKM'03)*. ACM, New York, NY, 116–123. DOI: http://dx.doi.org/10.1145/956863.956886

Jing Liu, Young-In Song, and Chin-Yew Lin. 2011. Competition-based user expertise score estimation. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'11)*. ACM, New York, NY, 425–434.

Mingrong Liu, Yicen Liu, and Qing Yang. 2010. Predicting best answerers for new questions in community question answering. In *Web-Age Information Management*. Lecture Notes in Computer Science, Vol. 6184. Springer, 127–138. DOI: http://dx.doi.org/10.1007/978-3-642-14246-8_15

Tie-Yan Liu. 2011. *Learning to Rank for Information Retrieval*. Springer.

Xiaoyong Liu, W. Bruce Croft, and Matthew B. Koll. 2005. Finding experts in community-based question-answering services. In *Proceedings of the 2005 ACM CIKM International Conference on Information and Knowledge Management*. ACM, New York, NY, 315–316.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013a. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 27th Annual Conference on Neural Information Processing Systems*. 3111–3119.

Tomas Mikolov, WenTau Yih, and Geoffrey Zweig. 2013b. Linguistic regularities in continuous space word representations. In *Proceedings of the Conference of the North American Chapter of the Association of Computational Linguistics*. 746–751.

Jeff Mitchell and Mirella Lapata. 2010. Composition in distributional models of semantics. *Cognitive Science* 34, 8, 1388–1429.

Ananth Mohan, Zheng Chen, and Kilian Q. Weinberger. 2011. Web-search ranking with initialized gradient boosted regression trees. In *Proceedings of the Yahoo! Learning to Rank Challenge*. 77–89.

Christof Monz. 2004. Minimal span weighting retrieval for question answering. In *Proceedings of the SIGIR 2004 Workshop on Information Retrieval for Question Answering*. 23–30.

Meredith Ringel Morris, Jaime Teevan, and Katrina Panovich. 2010. A comparison of information seeking using search engines and social networks. In *Proceedings of the 4th International AAAI Conference on Weblogs and Social Media*. 291–294.

Kevin Kyung Nam, Mark S. Ackerman, and Lada A. Adamic. 2009. Questions in, knowledge in? A study of Naver's question answering community. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'09)*. ACM, New York, NY, 779–788. DOI:http://dx.doi.org/10.1145/1518701.1518821

Michael G. Noll, Ching-Man Au Yeung, Nicholas Gibbins, Christoph Meinel, and Nigel Shadbolt. 2009. Telling experts from spammers: Expertise ranking in folksonomies. In *Proceedings of the 32nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'09)*. ACM, New York, NY, 612–619.

Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1999. *The PageRank Citation Ranking: Bringing Order to the Web.* Technical Report 1999-66. Stanford InfoLab, Stanford, CA.

Katrina Panovich, Rob Miller, and David R. Karger. 2012. Tie strength in question and answer on social network sites. In *Proceedings of the Conference on Computer Supported Cooperative Work (CSCW'12)*. ACM, New York, NY, 1057–1066.

Fatemeh Riahi, Zainab Zolaktaf, M. Mahdi Shafiei, and Evangelos E. Milios. 2012. Finding expert users in community question answering. In *Proceedings of the 21st International Conference on World Wide Web (WWW'12 Companion)*. 791–798.

Stefan Riezler, Alexander Vasserman, Ioannis Tsochantaridis, Vibhu O. Mittal, and Yi Liu. 2007. Statistical machine translation for query expansion in answer retrieval. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL'07)*.

Magnus Sahlgren. 2005. An introduction to random indexing. In *Proceedings of the Methods and Applications of Semantic Indexing Workshop at the 7th International Conference on Terminology and Knowledge Engineering*.

Magnus Sahlgren. 2006. *The Word-Space Model: Using Distributional Analysis to Represent Syntagmatic and Paradigmatic Relations Between Words in High-Dimensional Vector Spaces*. Ph.D. Dissertation. Stockholm University, Stockholm, Sweden.

G. M. Salton, A. Wong, and C. S. Yang. 1975. A vector space model for automatic indexing. *Communications of the ACM* 18, 11, 613–620.

Hinrich Schütze. 1998. Automatic word sense discrimination. *Computational Linguistics* 24, 1, 97–123.

Hinrich Schütze and Jan O. Pedersen. 1995. Information retrieval based on word senses. In *Proceedings of the 4th Annual Symposium on Document Analysis and Information Retrieval*. 161–175.

Linus Sellberg and Arne Jönsson. 2008. Using random indexing to improve singular value decomposition for latent semantic analysis. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC'08)*

Pavel Serdyukov and Djoerd Hiemstra. 2008. Modeling documents as mixtures of persons for expert finding. In *Advances in Information Retrieval*. Lecture Notes in Computer Science, Vol. 4956. Springer, 309–320.

Pavel Serdyukov, Henning Rode, and Djoerd Hiemstra. 2008. Modeling multi-step relevance propagation for expert finding. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management (CIKM'08)*. ACM, New York, NY, 1133–1142.

Aliaksei Severyn and Alessandro Moschitti. 2012. Structural relationships for large-scale learning of answer re-ranking. In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'12)*. 741–750.

Elena Smirnova and Krisztian Balog. 2011. A user-oriented model for expert finding. In *Advances in Information Retrieval*. Lecture Notes in Computer Science, Vol. 6611. Springer, 580–592.

Paul Smolensky. 1990. Tensor product variable binding and the representation of symbolic structures in connectionist systems. *Artificial Intelligence* 46, 1–2, 159–216.

Mark D. Smucker, James Allan, and Ben Carterette. 2007. A comparison of statistical significance tests for information retrieval evaluation. In *Proceedings of the 16th ACM Conference on Information and Knowledge Management (CIKM'07)*. 623–632.

Mihai Surdeanu, Massimiliano Ciaramita, and Hugo Zaragoza. 2011. Learning to rank answers to non-factoid questions from Web collections. *Computational Linguistics* 37, 2, 351–383.

Joseph P. Turian, Lev-Arie Ratinov, and Yoshua Bengio. 2010. Word representations: A simple and general method for semi-supervised learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL'10)*. 384–394.

Peter D. Turney. 2006. Similarity of semantic relations. *Computational Linguistics* 32, 3, 379–416.

Peter D. Turney and Michael L. Littman. 2005. Corpus-based learning of analogies and semantic relations. *Machine Learning* 60, 1–3, 251–278.

Peter D. Turney and Patrick Pantel. 2010. From frequency to meaning: Vector space models of semantics. *Journal of Artificial Intelligence Research* 37, 141–188.

Suzan Verberne, Lou Boves, Nelleke Oostdijk, and Peter-Arno Coppen. 2008. Using syntactic information for improving why-question answering. In *Proceedings of the 22nd International Conference on Computational Linguistics (COLING'08)*. 953–960.

Suzan Verberne, Lou Boves, Nelleke Oostdijk, and Peter-Arno Coppen. 2010. What is not in the bag of words for *Why*-QA? *Computational Linguistics* 36, 2, 229–245.

Suzan Verberne, Hans van Halteren, Daphne Theijssen, Stephan Raaijmakers, and Lou Boves. 2011. Learning to rank for *why*-question answering. *Information Retrieval* 14, 2, 107–132.

Dominic Widdows and Kathleen Ferraro. 2008. Semantic vectors: A scalable open source package and online technology management application. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC'08).*

Ludwig Wittgenstein. 1953. *Philosophische Untersuchungen*. Suhrkamp Verlag, Frankfurt am Main.

M. B. W. Wolfe, M. E. Schreiner, B. Rehder, D. Laham, P. W. Foltz, W. Kintsch, and T. K. Landauer. 1998. Learning from text: Matching readers and texts by latent semantic analysis. *Discourse Processes* 25, 2–3, 309–336.

Wen-Tau Yih, Ming-Wei Chang, Christopher Meek, and Andrzej Pastusiak. 2013. Question answering using enhanced lexical semantic models. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL'13).* 1744–1753.

Cheng-Xiang Zhai and John D. Lafferty. 2001. A study of smoothing methods for language models applied to ad hoc information retrieval. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'01)*. ACM, New York, NY, 334–342.

Jun Zhang, Mark S. Ackerman, and Lada Adamic. 2007a. CommunityNetSimulator: Using simulations to study online community networks. In *Communities and Technologies 2007*, C. Steinfield, B. T. Pentland, M. Ackerman, and N. Contractor (Eds.). Springer, London, UK, 295–321.

Jing Zhang, Jie Tang, and Juan-Zi Li. 2007b. Expert finding in a social network. In *Advances in Databases*. Lecture Notes in Computer Science, Vol. 4443. Springer, 1066–1069.

Guangyou Zhou, Yang Liu, Fang Liu, Daojian Zeng, and Jun Zhao. 2013. Improving question retrieval in community question answering using world knowledge. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI'13)*.

Tom Chao Zhou, Michael R. Lyu, and Irwin King. 2012. A classification-based approach to question routing in community question answering. In *Proceedings of the 21st International Conference on World Wide Web (WWW'12 Companion)*. 783–790.

Hengshu Zhu, Huanhuan Cao, Hui Xiong, Enhong Chen, and Jilei Tian. 2011. Towards expert finding by leveraging relevant categories in authority ranking. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management (CIKM'11)*. 2221–2224.

Received April 2016; accepted May 2016