A

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

Piero Molino, Universitá degli studi di Bari Aldo Moro, Italy
Luca Maria Aiello, Yahoo Labs Barcelona, Spain
Pasquale Lops, Universitá degli studi di Bari Aldo Moro, Italy

Community question answering sites expose a large amount of signals that have been leveraged for the tasks of best answer selection, question routing and detection of topical experts. Nevertheless, the vastness of the feature space has been explored only partially by individual research efforts, leaving a lack of a "big picture" on the features' effectiveness; additionally, several textual and expertise-network features suited for this domain are still unexplored or have never been used in combination. As a result, it is very difficult to assess which features are most effective and in which cases. Stemming from this observation, we gather under a learning-to-rank framework the most extensive feature set that has been used in the 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 Answer used for this task so far (40M answers), and we provide a faceted analysis of the results along different question categories. We propose a novel family of distributional semantics measures that can most of the times seamlessly replace widely used linguistic similarity features in the task of best answer prediction, being more than one order of magnitude faster to compute and providing greater predictive power. Also for the first time we combine textual features with novel network-based features for expert finding. The best feature set reaches an improvement between 21% and 27% in $P@1$ compared to three recent yet 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*

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

ACM Reference Format:
ACM Trans. Inf. Syst. V, N, Article A (YYYY), 36 pages.
DOI:http://dx.doi.org/10.1145/0000000.0000000

1. INTRODUCTION

Community Question Answering (CQA) sites such as Yahoo Answers, StackOverflow, or Ask.com have been very popular since the beginning of the rise of the Social Web due to their ability to leverage a collaborative paradigm to serve articulate user queries that may not find satisfactory answers if submitted to conventional search engines [Morris et al. 2010]. The management of the question-answering process in major online services is usually conducted by human users with very little support from automated techniques. In the last years, however, much research has been done to provide automated methods to identify topical experts, automatically routing to them questions relevant to their expertise and select the best answers among the ones provided (see §2 for an overview). In particular regarding the automatic best answer selection, which is a fundamental task in this field, an impressive amount of work has been conducted, but with a general lack of coherence in the experimental approaches. It is indeed difficult to establish how different methods or feature sets perform in relation to each others and in which cases.

In the general scenario of best answer prediction, there are some issues that we contribute to address with this work. First, there is a general lack of understanding of the relative predictive

DOI:http://dx.doi.org/10.1145/0000000.0000000

power of different textual and linguistic features that have been analyzed in isolation in several previous papers but have never been directly compared in an extensive, systematic way. Also, the space of linguistic features that are relevant to this task has still several untapped areas; in particular, in this work we explore a family of distributional semantics features. Second, textual approaches and expert-finding methods based on the networks of user interaction have been mainly considered independently and it is not clear how much the overall predictive potential can benefit from their integration. Last, the performance of the majority of best answer prediction methods in the literature has not been tested in relation to different question types or categories.

We contribute in this direction by considering a vast amount of lexical and language features, including semantic role labeling, machine translation and especially a novel class of distributional semantics features. We combine them, under a learning to rank approach, with a set of network-based features for expert finding that have always been considered in isolation. Our results have a two-fold implication. First, they allow us to shed light on the relative importance of features that have not been systematically compared in the past. Last, they led us to produce a supervised model with a predictive accuracy that tops three of the latest, yet already widely popular methods for best answer prediction.

More specifically, in this work we make the main following contributions.

— We consider 225 features in the CQA domain and we test them on the largest dataset from Yahoo Answers that has been ever used for this task. Some of those feature groups have been already considered in the literature but never compared directly to one another. The most effective combination of features reaches up to 27% performance gain on $P@1$ in best answer prediction over recent state of the art methods.
— We introduce new distributional semantics features. We verify that the information they convey is orthogonal to other well-established textual-based features, thus bringing a considerable additional power to prediction models. We find that those features can replace more costly and widely used textual similarity features without losing predictive performance in the general case.
— We propose, for the first time a combination of textual features with expertise network features in a learning to rank framework for the task of best answer prediction. We find that expertise networks information contributes significantly to the prediction results but far less than any other content signal.
— We break down the results by question type, finding that text quality features are more suited to predict the best answer to factual and subjective questions, whereas features from the user profile are more predictive for discussion and poll-type questions.

The paper is organized as follows. After a review of the related work (§2), we first describe in detail five main families of features that characterize most of CQA sites (§3). After describing the learning to rank framework we use to combine the features together (§4), the Yahoo Answers datasets we collected (§4.2), and the baselines (§4.3), we outline the experimental results in §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 §5.

2. RELATED WORK

Next, we discuss some background work in the field, describing studies on Community Question Answering and Expert Finding, as well as some of the features that are most widely adopted in previous approaches.

2.1. Community and non-Factoid Question Answering

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

One of the earliest and most widely known approaches to the best answer selection problem adopted different measures of text quality to find the best answer for a given question [Agichtein et al. 2008]. Intrinsic answer qualities such as grammatical, syntactic and semantic complexity, punctuation and typo errors are adopted, along with question-answer similarity and user expertise estimations. We build on that work by picking all 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 Non-Factoid QA systems, focusing on *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 Non-Factoid QA have been confirmed in several studies [Verberne et al. 2008; Verberne et al. 2010; Verberne et al. 2011], in which it is shown 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 Non-Factoid QA system. We also adopt distributional linguistic features adding even more levels of lexicalization to the linguistic representation.

Another set of approaches adopts machine translation 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 machine translation features, learning different translation models for different linguistic representations.

The study in the field dealing with the largest-scale dataset has been proposed by [Surdeanu et al. 2011]. They adopted 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 §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 been also 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 in 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], while 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].

Several slight variants to such approaches have been experimented during 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], as well as 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.

It is also worth mentioning that, in some cases, the task of expert finding has been addressed from a slightly different perspective that goes under the label of "question recommendation", that aims to recommend interesting questions for a contributor that is willing to provide answers. Such approaches tend to privilege the perspective of the answerer, for instance trying to assign questions also 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 [Dror et al. 2011] 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.

In 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. 2007b; 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 §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 network 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 beat also 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 the 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 recurring to 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 re-proposed, with more features and a more robust learning to rank algorithm, over StackOverflow data [Dalip et al. 2013], focusing on features specifically designed for that dataset, like 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 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 for a question. The first three families (*Text Quality*, *Linguistic Similarity* and *Distributional Semantics*) belong to the macro-group 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 user in answering a question by capturing either their historical information or their interactions with other members of the community. Next we give an overview for 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 measures quantitatively some properties of the text. The features belonging to this group count the number of whitespace violations 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 quoted parts of the answer 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]. [Full feature list in Tab. 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 multi-syllable words, and the presence of the words in whitelists. [Full feature list in Tab. IX].

Informativeness. This group of features was considered since a reasonable answer must contain some information that is not in the question, so we adopt 3 simple features that count the amount of nouns, verbs and adjectives occurring in the answer but not in the question. [Full feature list in Tab. 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 by different similarity and overlap measures. This part of our work follows the same approach.

Having analyzed 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. 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 we extract the n-grams. For the dependency parsing the chains are constructed in the form of *"dependant-relationType-head"* but we can extract also more general chains that do not contain the relationType. For the semantic role labeling, the chain has the form of *"predicate - argumentType - argument"*. Also 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 decide 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 candidate answer, we obtain the chains at different lexicalization level for both them and then apply a similarity metrics 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 we obtain *[be_guinness, guinness_kind, kind_beer]*. We do the same for the passage and we obtain *[guinness_produce, 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 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 the set of tokens belonging to the answer. With this simple overlap formula we calculate the overlap of unigrams with all the different lexical levels, resulting in 6 features. Other 15 features are obtained calculating the overlap of 2-grams, 3-grams and 4-grams of all the lexicalizations except for the named entities, as they are already n-grams of words in most of the cases. 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 8 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 6 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. [Full feature list in Tab. 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, to the BM25 weighting scheme and to the Language Modeling (with Dirichlet priors [Zhai and Lafferty 2001]) for all the different lexicalization levels except for the named entities, for a total of 15 features. [Full feature list in Tab. XII].

Density. We adopt a slight modification of the Minimal Span Weighting proposed by [Monz 2004], calculating it for all the different lexicalization levels. This yields 6 features. The original formula contains three components: text similarity, span size ratio and matching term ratio. The text similarity intercepts global similarity, the span intercepts local similarity and the matching term ratio counterbalances the local similarity, i.e. in the case of only one term matching out 5 question terms the span part of the formula would return a value of 1, while the matching term would be $\frac{1}{5}$. For such a reason, 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 have other features, like the frequency ones, that address for the text similarity, we retained only the local similarity part, resulting in the following formula:

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

where t_q and t_a are the sets of tokens respectively of the question and the answer; $\max(mms)$ and $\min(mms)$ are the initial and final location of the shortest sequence of answer tokens containing all the question tokens. [Full feature list in Tab. XIII]

Machine Translation. Research in machine translation, a sub-field 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 last decade, leading to better phrase-based translations. The objective of machine translation 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:

$$P(Q \mid A) = \prod_{q \in Q} P(q \mid A)$$

$$P(q \mid A) = (1-\lambda)P_{ml}(q \mid A) + \lambda P_{ml}(q \mid C)$$

$$P_{ml}(q \mid A) = \sum_{a \in A} \left(T(q \mid a) P_{ml}(q \mid A)\right)$$

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 re-scale $P(w'|w)$ for all the 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 the lexicalization degrees and for all the combinations of dependencies and semantic role labeling chains, we obtain 14 features. [Full feature list in Tab. XIV].

Others. We consider 4 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. [Full feature list in Tab. XV].

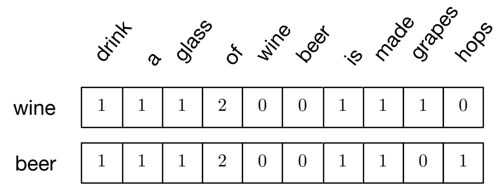

Fig. 1. An example of vector representation of words in a DSM

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 of 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.

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 word 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 root in the Vector Space Model [Salton et al. 1975], one 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 VSMs for semantic processing of text has been done by Turney et al. [2010]; an analysis of some compositional operators is described in the work by Mitchell et al. [Mitchell and Lapata 2010]. Naturally, also several applications to IR exist [2008], including term-doc matrix reduction [Deerwester et al. 1990] and ambiguity resolution [Schütze and Pedersen 1995; Basile et al. 2011].

DSMs have not been used in any task directly related to CQAs, so far. 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 we adopt and finally we describe our strategy to integrate it inside our best answer predictor.

	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. An example of $term \times term$ matrix $\mathbf{M}$

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[1] and its vocabulary V, a $n \times n$ 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 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 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 Skipgram Models. All 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, while k is the number of reduced dimensions.

Latent Semantic Analysis. Latent Semantic Analysis [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, while 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 $\widetilde{\mathbf{M}}$ s 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

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

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

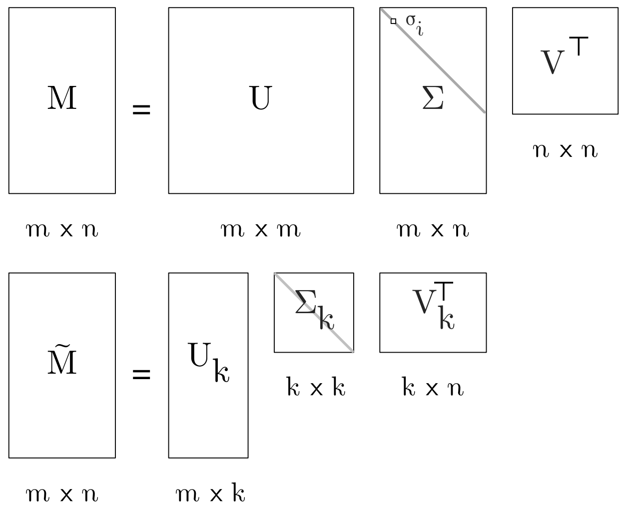

Fig. 3. A depiction of SVD matrices

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 Random Indexing (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 to solve efficiently 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.

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:

$$\mathbf{M'}_{n,k} = \mathbf{M}_{n,m}\mathbf{R}_{m,k} \quad k << m \tag{1}$$

The new matrix $\mathbf{M}'$ has the property to preserve the distance between points. This property is known as 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 $d_r = c \cdot d$. A proof of that property has been done by Dasgupta et al. [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 non-zero 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

Dataset

I drink **beer**

You drink a glass of **beer**

Context Vectors

I	1	0	0	0	0	-1	0
drink	0	0	1	0	0	0	0
beer	0	1	0	0	0	0	0
you	0	-1	0	0	0	0	1
glass	-1	0	0	0	1	0	0

Term Vectors

$$tv_{\text{beer}} = 1cv_{i} + 2cv_{\text{drink}} + 1cv_{\text{you}} + 1cv_{\text{glass}}$$

beer	0	-1	2	0	1	-1	1

Fig. 4. Term Vector construction in Random Indexing

Latent Semantic Analysis over Random Indexing. Computing LSA on the co-occurrence matrix **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 **M** produced by Random Indexing. Sellberg and Jonsson [2008] followed a similar approach for the retrieval of similar FAQs in a QA system. Their experiments 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 DSMs 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 data sets.

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 non-linear hidden layers [Bengio et al. 2003]. Two such simpler networks are proposed in that work: the Continuous Bag-of-Words Model and Continuous Skip-gram Model. While both are shown to be effective in semantic-syntactic word relationship learning and sentence completion tasks, the former is faster to train, while the latter has better performances at the cost of slightly longer training time. Although 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 non-linear hidden layer, this improves the learning efficiency at the expenses of a representation that might be less precise, but enables to learn models with bigger amounts of data. The model, shown in Figure 5, iterates over the words in the dataset and uses each

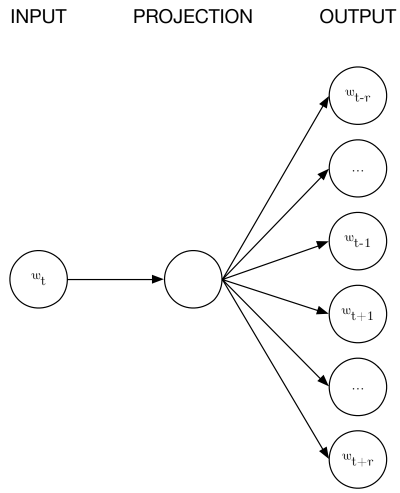

Fig. 5. The architecture of the Continuous Skip-gram Model

word w_t as an input to a log-linear classifier with continuous projection layer. What it outputs is a prediction of the words within a certain range before and after the input word.

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}.

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.

Distributional Semantic Models Integration in Question Answering. We now discuss how to leverage the word vector representations to match questions to the best answers. We use word vector representations for building 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 we 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:

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

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 point-wise sum of components:

$$\mathbf{s} = \mathbf{u} + \mathbf{v} \tag{3}$$

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].

Given a phrase or sentence s, we denote with **s** its vector representation obtained applying addition operator $(+)$ to the vector representation of terms it is composed of. Furthermore, it is possible to compute the similarity between two phrases / sentences exploiting the cosine similarity between vectors (2).

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 **q** and **a** which represent respectively the question and the candidate answer in a semantic space. Vector representations for question and answer are built applying the addition operator to the vector representation of words belonging to them:

$$\begin{aligned} \mathbf{q} &= q_1 + q_2 + \ldots + q_n \\ \mathbf{a} &= a_1 + a_2 + \ldots + a_m \end{aligned} \tag{4}$$

The similarity between **q** and **a** is computed as the cosine similarity between them.

For example, we want to compare the question **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 (non-stopword) words are:

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

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

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

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

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

$$\begin{aligned} &\cos(\mathbf{q}, \mathbf{a^1}) = 0.9846 \\ &\cos(\mathbf{q}, \mathbf{a^2}) = 0.7794 \end{aligned}$$

So $\mathbf{a^1}$ 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 **M** matrix both using Wikipedia as a corpus and using the set of all the answers in the training set obtained from the *Yahoo Answers 2011 dataset* that we use for the evaluation (see §4.2 and §4.4). We do so to use both general purpose texts incorporating common sense knowledge and knowledge that is specific to the dataset we want to actually use. The number of dimensions of the vector representations for all the 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 the corpora, resulting in 8 features. [Full set of features in Tab. 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 systems or allow the possibility of collaborative tagging. Although most of the features 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 sub-groups of user features are summarized next.

User Profile. The user profile contain information that might be a good proxy for the level of user's involvement in the community. These include: the presence of a resume, of a textual self-description of the user, of a title and a 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 5 features. [Full set of features in Tab. XVII].

Question and Answers. The number of questions the user asked, deleted, answered, flagged, 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 the questions asked. We replicate the same features we calculated on the questions asked by the user also on the answers given by the user, adding also 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. [Full set of features in Tab. XVIII].

Question categories. We replicate the same features defined for the question and answer history of the user, but considering only 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 the asker is interested in. So we add additional 19 features for questions in the category and other 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 the different categories ($\|p\|$ is the number of categories).

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

This allows us to evaluate how specific (high entropy) or spread out (low entropy) the user knowledge (or interest) is. [Full set of features in Tab. 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, 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 8 features. [Full set of features in Tab. XX].

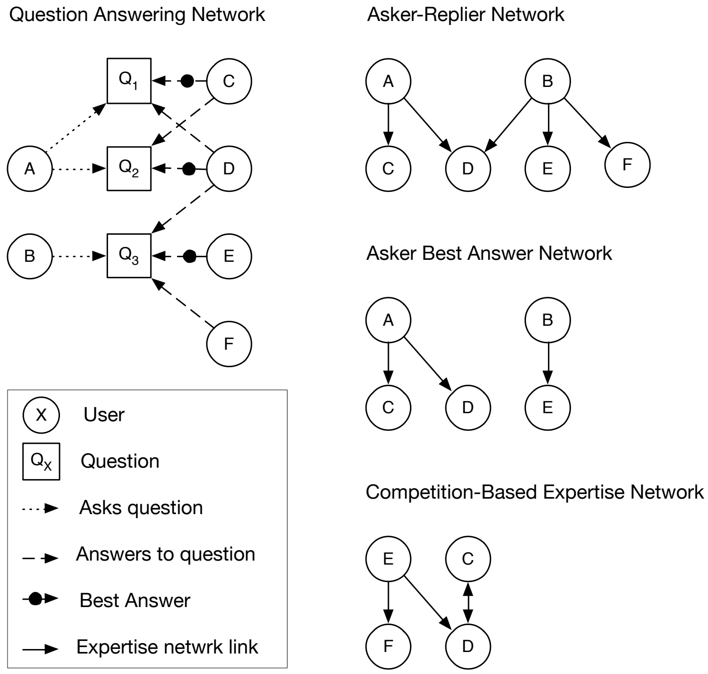

Fig. 6. The graph of relations between askers, questions, and answerers (left) and the three types of Expertise Networks derived by it (right).

3.5. Expertise Network Features (n)

The network features we propose arise from expert finding literature, where a content-agnostic analysis of the interactions between participants in CQA is shown 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. Also, the estimation of the users' expertise may not depend just 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, answering correctly a question made by an expert is a better indication of expertise than answering a question made by a newbie.

These considerations has 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 have been defined and studied for CQA. We provide visual examples for each in Figure 6.

The first is the *Asker Replier Network (ARN)* [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 the users who answered the same question towards 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 the answers to have equal value. ABAN addresses this problem but 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. Also, 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 question answering communities.

The application of graph *centrality* metrics to the Expertise Networks mentioned above 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, but in all the 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 we aim to include in a learning to rank framework a wide set of features, therefore we do not restrict ourselves to specific pairs but we consider instead all the 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 9 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. [Full set of features in Tab. XXI]

4. EXPERIMENTAL EVALUATION

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

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 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 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, but 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 datapoint and a set of datapoints 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 Random Forests (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 a 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 $y_i \in Y$.

The RF algorithm trains a model H such that $H(x_i) \approx y_i$ and so that the ranking of all the 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: $\quad D_t \leftarrow sample(D)$
3: $\quad K \leftarrow roandomPick(m)$
4: $\quad h_i \leftarrow buildDecisionTree(D_t, K)$
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 decision tree regression algorithm 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 decision 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 the 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 decision trees prevent 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[2] with the default parameters.

4.2. Dataset

The instance of CQA 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 (TLC), such as Politics, Sports or Entertainment, and a growing number of Leaf-Level Categories (LLC) –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 with optional detailed description, and a mandatory leaf-level category that is assigned by the asker. Questions have a lifecycle of states that goes from *open*, to *voting* and finally to *resolved*, and users can actively moderate content using several feedback mechanisms, such 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 the 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 with majority vote. The process of best answer selection is important not only to reward contributors according to the Yahoo Answers incentive scheme[3], 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 best answer assigned by the asker, $>$ 39.5M answers and $>$ 6.1M unique users. 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

[2]http://sourceforge.net/p/lemur/wiki/RankLib/
[3]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 available at: http://answers.yahoo.com/info/scoring_system

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 it covers 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 leaf level categories using features such as the average number of replies to a question and the average 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 freely available dataset [4] that contain a sample of manner questions collected from the US 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 the 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 IR 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, along with 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, as the dataset they adopted for the experiments was made of 8,366 question/answer pairs so their results were not directly comparable with ours. We chose this baseline because it was the state of the art on Yahoo Answers data.
— *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 *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 a SVM-based ranker. Their evaluation was carried out on *Yahoo Answers Manner*

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

Questions dataset as well. We choose this baseline because it was the state of the art on *Yahoo Answers Manner Questions dataset* for MRR.

Features	**MRR**	**P@1**	**DCG**
BM25	0.5532	0.4143	0.6567
[Agichtein et al. 2008]	0.6375	0.5243	0.6962
tq	0.7016	0.5305	0.7655
ls	0.6921	0.5143	0.7613
ds	0.6760	0.4782	0.7564
u	0.7009	0.5218	0.7757
n	0.6645	0.4527	0.7484
tq+u	0.7597	0.6201	0.8260
tq+n	0.7366	0.5862	0.8080
tq+ds	0.7144	0.5536	0.7910
tq+ls	0.7129	0.5515	0.7897
tq+u+n	0.7742	0.6416	0.8370
tq+u+ds	0.7606	0.6210	0.8266
tq+u+ls	0.7597	0.6199	0.8260
tq+lo+ds	0.7143	0.5519	0.7901
tq+u+n+ds	0.7752	0.6450	0.8379
tq+u+n+ls	0.7739	0.6414	0.8368
all	0.7798	0.6471	0.8389

Table I: Predictive power of the learning to rank framework trained on different feature subsets, on the *Yahoo Answers 2011 dataset*. 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**.

4.4. Performance analysis

We evaluate our learning to rank framework splitting the dataset into 70% training, 10% development and 20% test, using a temporal criterion (older questions for training). The 10% development was needed for tuning machine translation parameter λ (see §3.2). For each question, all 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 IR metrics that have been commonly used to evaluate this task, namely *Mean Reciprocal Rank (MRR)*, *Precision at 1 (P@1)* and *Discounted Cumulative Gain (DCG)*. When considering the answers to a single question, these are formally defined as follows:

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

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 the scores are then averaged over all the 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. All differences have to be considered statistically significant (using the non-parametric Randomization test, as suggested by Smucker et al. [2007], with $p < 0.01$) unless otherwise specified.

Feature	Δ
tq: Preposition Count	0.049
tq: Verbs not in Question	0.045
tq: Nouns not in Question	0.045
tq: Unique Words in Answer	0.043
tq: Pronouns Count	0.042
tq: Punctuation Count	0.039
tq: Average Words per Sentence	0.039
ds: Random Indexing on Yahoo Answers	0.039
ls: Super-senses Overlap	0.038
tq: Adjectives not in Question	0.036
tq: Conjunctions Count	0.035
tq: Capitalized Words Count	0.035
tq: "To be" Count	0.035
ls: Lemma Overlap	0.034
ls: Stem Overlap	0.034
ls: Term Overlap	0.032
tq: Auxiliary Verbs Count	0.034
ls: Super-senses BM25	0.031
n: Indegree on CBEN	0.030
u: Answerer's Best Answer Ratio	0.030

Table II: Ablation test. 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.

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 the others to select the best combination. The process is repeated until all the 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], that found simple user features such as the percentage of best answers very predictive of the level of 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 6 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] has also 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 we consider inside each family.

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 they bring is very overlapping. Indeed, their combination is the worst performing among all the feature pairings. The same happens with **ds** features. **n**, and especially **u** features, are instead

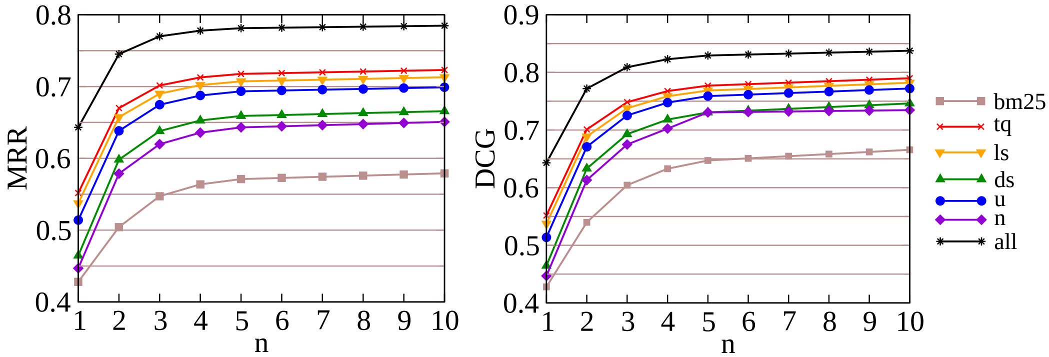

Fig. 7. MRR and DCG computed for the first n positions of the ranking, for the different features families, plus the BM25 baseline and the full set of features.

more orthogonal to the **tq** information and are able to boost the performance considerably. Most importantly, we find that **n** and **u** features carry predictive information that is non-overlapping, 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 more viable alternative.

The *MRR* score obtained with the combination of all the feature groups is a **22%** improvement over the baseline, while the *P@1* score is a **23%** improvement and *DCG* score is a **21%** 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 (8^{th} and 19^{th}, respectively).

As final remark, we note that when plotting the MRR and DCG for rankings that include the top n results only (Figure 7) 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 2 or 3 answers.

4.4.2. Performance on Yahoo Answers Manner Questions. The last two baselines we consider ([Surdeanu et al. 2011] and [Hieber and Riezler 2011]) have been applied to the smaller *Yahoo Answers Manner Questions dataset* described in §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 before. A Random Forest model is learned for each feature set, performances are reported in Table III. Differently from the previous dataset, we do not perform a temporal split as timestamps are not available in the dataset, so we perform 5-fold cross validation with a 70-10-20 split (10% being a validation set).

All the 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**.

Features	P@1	MRR	DCG
BM25	0.4112	0.5606	0.6121
[Surdeanu et al. 2011]	0.5091	0.6465	-
[Hieber and Riezler 2011]	0.4844	0.6676	-
ds	0.6118	0.7689	0.8198
ls	0.618	0.7717	0.8236
tq	0.6245	0.7857	0.8352
ds+ls	0.618	0.7721	0.8236
ds+tq	0.6532	0.7920	0.8421
ls+tq	0.6401	0.7855	0.8352
ds+ls+tq	0.6532	0.7922	0.8425

Table III: Predictive power of the learning to rank framework trained on different feature subsets, on the *Yahoo Answers Manner Questions dataset*.

Feature	Rank
ds: Random Indexing on Yahoo Answers	8
ds: Continuous Skip-gram Model on Yahoo Answers	30
ds: LSA on Wikipedia	37
ds: LSA after Random Indexing on Wikipedia	38
ds: Continuous Skip-gram Model on Wikipedia	39
ds: Random Indexing on Wikipedia	40
ds: LSA after Random Indexing on Yahoo Answers	89
ds: LSA on Yahoo Answers	90

Table IV: Distributional-Semantics-based features ablation ranking

Taking into account the combinations of features we observe that the best performing one is the composition of **ds** and **tq**. The combinations of **ds** and **ls** does not improve at all for *P@1* and improves just of 0.004 for *MRR* over the **ls** group alone, a non statistically 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** does not improve the performances at all for *P@1* and improves just of 0.002 for **MRR** and 0.004 for **DCG**, again a non statistically significant improvement. This finding suggests that in this setting the linguistic features, that requires a really expensive preprocessing time to be computed, can be substituted with a single features 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 **27%** improvement over the state of the art (best of the three baselines), while the best *MRR* scores obtained with the **ds+tq+ls** features group are an improvement of **18%** 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.

Considering the features based on Distributional Semantics (**ds**), reported in Table IV, we can clearly see that the best performing feature, Random Indexing on Yahoo Answers, ranks 8^{th}. This is encouraging and suggests that the adoption of textual data coming from the dataset itself is helpful. Continuous Skip-gram Model on the same datasets is the second best one, ranking 30^{th}, supporting the suggestion of the Random Indexing feature. The other two features using models learned on the same dataset rank 89^{th} (LSA over Random Indexing) and 90^{th} (LSA), almost in the middle of the ranking. The difference with respect to Random Indexing suggests that probably the number of

Feature	Rank
n: Indegree on CBEN	19
n: Hits on CBEN	32
n: Indegree on ABAN	101
n: Hits on ABAN	108
n: Indegree on ARN	161
n: Hits on ARN	164
n: PageRank on ARN	170
n: PageRank on CBEN	183
n: PageRank on ABAN	184

Table V: Network features ablation ranking

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: 37^{th} for LSA, 38^{th} for LSA over Random Indexing, 39^{th} for Continuous Skip-gram Model and 40^{th} for Random Indexing. This suggests that the differences in models, in this case, are less influent than the dataset itself. As Wikipedia contains more than 4 million articles, the huge quantity of text in this dataset leads to similar behaving models.

Considering the Network based features (**n**), reported in Table V, the best performing network structure is the Competition-Based Expertise Networks. Two features based on models calculated on this network are the top ranked: Indegree on CBEN is 19^{th} and Hits on CBEN is 32^{nd}. The same two models calculated on the Asker Best-Answerer Network are ranked in the middle of the ranking, 101-st and 108-th respectively, while those calculated on the Asker Replier Network are ranked lower in the ranking, 161^{th} and 164^{th}. 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 really similar way. At the bottom of the ranking we found the PageRank model calculated on ARN (170^{th}), on CBEN (183^{rd}) and ABAN (184^{th}). 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 "high-quality" answer. To investigate this aspect, we get back to the bigger *Yahoo Answers 2011 dataset* and we break down the performance of the different feature families by the four question categories we defined in §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, while the **u** features are dominating discussions and polls.

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

4.4.5. Different Algorithms. Our decision to use a pointwise approach like RF as 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.

	Factual	**Subjective**	**Discussion**	**Poll**
tq	**0.7329**	**0.7242**	0.6676	0.6762
ls	0.7243	0.7117	0.6482	0.6350
ds	0.6879	0.6738	0.6377	0.6498
u	0.7221	0.7118	**0.6724**	**0.6878**
n	0.7003	0.6953	0.6132	0.6214
all	0.8059	0.7898	0.7508	0.7644

Table VI: MRR scores obtained with single feature families on the *Yahoo Answers 2011 dataset*.

	LR	**RankSVM**	**ListNet**	**RF**
Manner	0.6952	0.7683	0.7520	**0.7922**
Factual	0.7407	0.7774	0.7626	**0.8059**
Subjective	0.7183	0.7640	0.7411	**0.7898**
Discussion	0.6881	0.7256	0.7059	**0.7508**
Poll	0.7027	0.7286	0.7312	**0.7644**
All	0.7165	0.7491	0.7466	**0.7798**

Table VII: MRR scores obtained with different Learning to Rank algorithms on the *Yahoo Answers 2011 dataset*.

Moreover, due to the large size of the dataset, we believed that the resilience to overfitting that characterize RF would have been a great advantage.

To asses in our intuition was likely to be true, we run the evaluations on the same datasets with the same features, but using different algorithms.

We chose Logistic Regression (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]. All the algorithms are used with their default parameters from the adopted libraries (RankLib[5] and SVMLight[6]), without a specific hyperparameter tuning.

The results in Table VII show only the trends for MRR using all the features, but the same trends are also present by changing the adopted feature set combination and metric. Logistic Regression is the worst performing algorithm on all the sets of questions, while among RankSVM and ListNet the difference is really small with RankSVM obtaining slightly higher results on all question sets but *Poll*. None of the alternative algorithms can 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. Besides being able to outperform the prediction ability of state-of-the-art methods up to 27% in P@1, our experiments allow us also to

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

draw important conclusions about the impact of different features employed that have never been spell 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 due to the fact that the content of the question and answers (their topic and structure) are a more important source of information to determine the question-answering match rather than the expertise of the answerers. Those features are to prefer when dealing with factual-type questions.
— Among the textual features, *Text Quality and Distributional Semantics are in general to prefer 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 a number of previous approaches but are roughly 12 times more computationally expensive than Distributional Semantics ones.
— *The new Distributional-Semantics-based approach 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 them in combination with text quality features instead of using different textual features combined.

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

Acknowledgments

We are grateful to Çiğdem Aslay for her precious suggestions and support in gathering the data. This research is partially supported by 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 (www.cenitsocialmedia.es) "Social Media", and by Grant TIN2009-14560-C03-01of the Ministry of Science and Innovation of Spain.

Appendix

The detailed list of features, breakdown by types, is reported next.

Group tq	**Subgroup** visual property
Count of auxiliary verb	
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 over all characters	
Whitespace characters over all characters	
Capital letters characters over all characters	

Table VIII: Visual Property features

Group tq	**Subgroup** readability
Average words per sentence	
Average words length in syllables	
Average words length in characters	
Number of complex words over 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 IX: Readability 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 X: Informativeness 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 bigams
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 trigams
Overlap of term trigrams
Overlap of lemma tetragram
Overlap of tetragrams of concatenations of lemmas and PoS tags
Overlap of stem tetragrams
Overlap of super-sense tetragams
Overlap of term tetragrams

Table XI: Overlap 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 XII: Frequency 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 XIII: Density 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 XIV: Machine Translation features

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

1 over the length of the question

Table XV: Other features

Group ls	**Subgroup** distributional semantics
Semantic similarity using LSA on Wikipedia corpus	
Semantic similarity using Random Indexing on Wikipedia corpus	
Semantic similarity using LSA after Random Indexing on Wikipedia corpus	
Semantic similarity using Continuous Skip-gram Model on Wikipedia corpus	
Semantic similarity using LSA on Yahoo Answers corpus	
Semantic similarity using Random Indexing on Yahoo Answers corpus	
Semantic similarity using LSA after Random Indexing on Yahoo Answers corpus	
Semantic similarity using Continuous Skip-gram Model on Yahoo Answers corpus	

Table XVI: Distributional-Semantics-based 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 XVII: User Profile 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	
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 over all the questions asked by the user	
Number of deleted questions over all the questions of the user	
Number of answered questions over all the questions asked by the user	
Number of flagged questions over all the questions asked by the user	
Number of questions with a star over all the 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 over number of answers given by the user	
Number of (non 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
Number of deleted answers given by the user
Number of best answers given by the user
Number of flagged answers given by the user
Number of (not deleted) answers over all the answers given by the user
Number of deleted answers over all the answers given by the user
Number of best answers over all the answers given by the user
Number of flagged answers over all the 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 between pos. and neg. votes that the user's answers have received
Ratio between pos. and neg. votes 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 XVIII: Question Answer features

Group u | **Subgroup** category
Number of (not deleted) questions asked by the user in the question's category
Number of deleted questions asked by the user in the question's category
Number of answered questions asked by the user in the question's category
Number of flagged questions asked by the user in the question's category
Number of questions with a star asked by the user in the question's category
Number of (not deleted) questions asked by the user in the question's category
Number of deleted questions asked by the user in the question's category
Number of answered questions asked by the user in the question's category
Number of flagged questions asked by the user in the question's category
Number of questions with a star asked by the user in the question's category
Portion of (not deleted) questions asked by the user in the question's category
Portion of deleted questions of the user in the question's category
Portion of user's asked questions answered in the question's category
Portion of user's questions flagged in the question's category
Portion of user's questions starred in the question's category
Minimum Automatic Readability of user's questions in the question's category
Maximum Automatic Readability of user's questions in the question's category
Average Automatic Readability of user's questions in the question's category
Ratio between no. of questions and answers of the user in the question's category
Number of (non deleted) answers given by the user in the question's category
Number of deleted answers given by the user in the question's category
Number of best answers given by the user in the question's category
Number of flagged questions asked by the user in the question's category
Number of (not deleted) answers given by the user in the question's category
Number of deleted answers given by the user in the question's category
Number of best answers given by the user in the question's category
Number of flagged answers given by the user in the question's category
Portion of (not deleted) answers given by the user in the question's category
Portion of deleted answers among those given by the user in the question's category
Portion of best answers among those given by the user in the question's category
Portion of flagged answers among those given by the user in the question's category
Number of positive votes for the answers given by the user in the question's category

Number of negative votes for the user's answers in the question's category
Difference between pos. and neg. votes for user's answers in the question's category
Ratio between pos. and neg. votes for user's answers in the question's category
Minimum Automatic Readability Index of user's answers in the question's category
Maximum Automatic Readability Index of user's answers in the question's category
Average Automatic Readability Index of user's answers in the question's category
Entropy of the vector of the number of questions in each category
Entropy of the vector of the number of answers in each category
Entropy of the vector of the number of questions and answers in each category

Table XIX: Category 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 number of positive and negative votes given by the user	
Number of positive votes over 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 XX: Behavioral features

Group n	**Subgroup** arn - aban - cben
Indegree of the user in the Asker Replier Network	
PageRank of the user in the Asker Replier Network	
Hits Authority of the user in the Asker Replier Network	
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 Competition-Based Expertise Network	
PageRank of the user in the Competition-Based Expertise Network	
Hits Authority of the user in the Competition-Based Expertise Network	

Table XXI: Network features

REFERENCES

2008. *Proceedings of the International Conference on Language Resources and Evaluation, LREC 2008, 26 May - 1 June 2008, Marrakech, Morocco*. European Language Resources Association.

Lada A. Adamic, Jun Zhang, Eytan Bakshy, and Mark S. Ackerman. 2008. Knowledge sharing and yahoo answers: everyone knows something, See Huai et al. [2008], 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 Web Data Mining, WSDM 2008, Palo Alto, California, USA, February 11-12, 2008*, Marc Najork, Andrei Z. Broder, and Soumen Chakrabarti (Eds.). ACM, 183–194.

Çigdem Aslay, Neil O'Hare, Luca Maria Aiello, and Alejandro Jaimes. 2013. Competition-based networks for expert finding, See Jones et al. [2013], 1033–1036.

Krisztian Balog, Leif Azzopardi, and Maarten de Rijke. 2006. Formal models for expert finding in enterprise corpora. In *SIGIR 2006: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in*

Information Retrieval, Seattle, Washington, USA, August 6-11, 2006, Efthimis N. Efthimiadis, Susan T. Dumais, David Hawking, and Kalervo Järvelin (Eds.). ACM, 43–50.

Pierpaolo Basile. 2011. Super-Sense Tagging Using Support Vector Machines and Distributional Features. In *Evaluation of Natural Language and Speech Tools for Italian, International Workshop, EVALITA 2011, Rome, Italy, January 24-25, 2012, Revised Selected Papers (Lecture Notes in Computer Science)*, Bernardo Magnini, Francesco Cutugno, Mauro Falcone, and Emanuele Pianta (Eds.), 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*, Alessandro Soro, Eloisa Vargiu, Giuliano Armano, and Gavino Paddeu (Eds.). Studies in Computational Intelligence, Vol. 324. Springer Berlin / Heidelberg, 249–265.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. 2003. A Neural Probabilistic Language Model. *Journal of Machine Learning Research* 3 (2003), 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 *SIGIR*. 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, See Huai et al. [2008], 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 2009, Madrid, Spain, April 20-24, 2009*, Juan Quemada, Gonzalo León, Yoëlle S. Maarek, and Wolfgang Nejdl (Eds.). ACM, 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, See Huang et al. [2010], 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, Las Vegas, Nevada, USA, August 24-27, 2008*, Ying Li, Bing Liu, and Sunita Sarawagi (Eds.). ACM, 866–874.

Alessandro Bozzon, Marco Brambilla, Stefano Ceri, Matteo Silvestri, and Giuliano Vesci. 2013. Choosing the right crowd: expert finding in social networks. In *Joint 2013 EDBT/ICDT Conferences, EDBT '13 Proceedings, Genoa, Italy, March 18-22, 2013*, Giovanna Guerrini and Norman W. Paton (Eds.). ACM, 637–648.

Leo Breiman. 2001. Random Forests. *Machine Learning* 45, 1 (2001), 5–32.

Curt Burgess, Kay Livesay, and Kevin Lund. 1998. Explorations in context space: Words, sentences, discourse. *Discourse Processes* 25, 2-3 (1998), 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, New Orleans, Louisiana, USA, November 2-8, 2003*. ACM, 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 *Machine Learning, Proceedings of the Twenty-Fourth International Conference (ICML 2007), Corvallis, Oregon, USA, June 20-24, 2007 (ACM International Conference Proceeding Series)*, Zoubin Ghahramani (Ed.), Vol. 227. ACM, 129–136.

Shuo Chang and Aditya Pal. 2013. Routing questions for collaborative answering in community question answering. In *Advances in Social Networks Analysis and Mining 2013, ASONAM '13, Niagara, ON, Canada - August 25 - 29, 2013*, Jon G. Rokne and Christos Faloutsos (Eds.). ACM, 494–501.

Bee-Chung Chen, Anirban Dasgupta, Xuanhui Wang, and Jie Yang. 2012. Vote calibration in community question-answering systems, See Hersh et al. [2012], 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 Fifteenth Text REtrieval Conference, TREC 2006, Gaithersburg, Maryland, November 14-17, 2006*, Ellen M. Voorhees and Lori P. Buckland (Eds.), Vol. Special Publication 500-272. National Institute of Standards and Technology (NIST).

Lin Chen and Richi Nayak. 2008. Expertise Analysis in a Question Answer Portal for Author Ranking. In *2008 IEEE / WIC / ACM International Conference on Web Intelligence, WI 2008, 9-12 December 2008, Sydney, NSW, Australia, Main Conference Proceedings*. IEEE Computer Society, 134–140.

William W. Cohen and Samuel Gosling (Eds.). 2010. *Proceedings of the Fourth International Conference on Weblogs and Social Media, ICWSM 2010, Washington, DC, USA, May 23-26, 2010*. The AAAI Press.

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 (2011), 2493–2537.

Nick Craswell, David Hawking, Anne-Marie Vercoustre, and Wilkins Peter. 2001. Panoptic Expert: Searching for experts not just for documents. In *Ausweb 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, See Jones et al. [2013], 543–552.

S. Dasgupta and A. Gupta. 1999. *An elementary proof of the Johnson-Lindenstrauss lemma*. Technical Report. Technical Report TR-99-006, International Computer Science Institute, Berkeley, California, USA.

Scott C. Deerwester, Susan T. Dumais, Thomas K. Landauer, George W. Furnas, and Richard A. Harshman. 1990. Indexing by Latent Semantic Analysis. *JASIS* 41, 6 (1990), 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 2003, San Diego, California, USA, June 13, 2003*, Mohammed Javeed Zaki and Charu C. Aggarwal (Eds.). ACM, 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, San Diego, CA, USA, August 21-24, 2011*, Chid Apté, Joydeep Ghosh, and Padhraic Smyth (Eds.). ACM, 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, 7-12 July 2003, Sapporo Convention Center, Sapporo, Japan.*, Erhard W. Hinrichs and Dan Roth (Eds.). ACL, 16–23.

David A. Ferrucci. 2011. IBM's Watson/DeepQA. *SIGARCH Computer Architecture News* 39, 3 (2011).

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 (1999).

Yupeng Fu, Rongjing Xiang, Yiqun Liu, Min Zhang, and Shaoping Ma. 2007. Finding Experts Using Social Network Analysis. In *2007 IEEE / WIC / ACM International Conference on Web Intelligence, WI 2007, 2-5 November 2007, Silicon Valley, CA, USA, Main Conference Proceedings*. IEEE Computer Society, 77–80.

Daniel Gildea and Daniel Jurafsky. 2002. Automatic Labeling of Semantic Roles. *Computational Linguistics* 28, 3 (2002), 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 2013 San Francisco, CA, USA, October 28, 2013*, Jalal Mahmud, James Caverlee, Jeffrey Nichols, John O'Donovan, and Michelle X. Zhou (Eds.). ACM, 25–28.

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

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 2009, Boston, MA, USA, April 4-9, 2009*, Dan R. Jr. Olsen, Richard B. Arthur, Ken Hinckley, Meredith Ringel Morris, Scott E. Hudson, and Saul Greenberg (Eds.). ACM, 759–768.

William R. Hersh, Jamie Callan, Yoelle Maarek, and Mark Sanderson (Eds.). 2012. *The 35th International ACM SIGIR conference on research and development in Information Retrieval, SIGIR '12, Portland, OR, USA, August 12-16, 2012*. ACM.

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 2011, Glasgow, United Kingdom, October 28, 2011*, Iván Cantador, Francisco M. Carrero, José Carlos Cortizo, Paolo Rosso, Markus Schedl, and José A. Troyano (Eds.). ACM, 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 2010, Raleigh, North Carolina, USA, April 26-30, 2010*, Michael Rappa, Paul Jones, Juliana Freire, and Soumen Chakrabarti (Eds.). ACM, 431–440.

Jinpeng Huai, Robin Chen, Hsiao-Wuen Hon, Yunhao Liu, Wei-Ying Ma, Andrew Tomkins, and Xiaodong Zhang (Eds.). 2008. *Proceedings of the 17th International Conference on World Wide Web, WWW 2008, Beijing, China, April 21-25, 2008*. ACM.

Jimmy Huang, Nick Koudas, Gareth J. F. Jones, Xindong Wu, Kevyn Collins-Thompson, and Aijun An (Eds.). 2010. *Proceedings of the 19th ACM Conference on Information and Knowledge Management, CIKM 2010, Toronto, Ontario, Canada, October 26-30, 2010*. ACM.

Thorsten Joachims. 2002. Optimizing search engines using clickthrough data. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, July 23-26, 2002, Edmonton, Alberta, Canada*. ACM, 133–142.

William B. Johnson and Joram Lindenstrauss. 1984. Extensions of Lipschitz mappings into a Hilbert space. *Conference on Modern Analysis and Probability, Contemporary Mathematics* 26 (1984), 189–206.

Gareth J. F. Jones, Paraic Sheridan, Diane Kelly, Maarten de Rijke, and Tetsuya Sakai (Eds.). 2013. *The 36th International ACM SIGIR conference on research and development in Information Retrieval, SIGIR '13, Dublin, Ireland - July 28 - August 01, 2013*. ACM.

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

Pawel Jurczyk and Eugene Agichtein. 2007. Discovering authorities in question answer communities by using link analysis, See Silva et al. [2007], 919–922.

Yutaka Kabutoya, Tomoharu Iwata, Hisako Shiohara, and Ko Fujimura. 2010. Effective Question Recommendation Based on Multiple Features for Question Answering Communities, See Cohen and Gosling [2010].

Pentti Kanerva. 1988. *Sparse Distributed Memory*. MIT Press.

Wei-Chen Kao, Duen-Ren Liu, and Shiu-Wen Wang. 2010. Expert finding in question-answering websites: a novel hybrid approach. In *Proceedings of the 2010 ACM Symposium on Applied Computing (SAC), Sierre, Switzerland, March 22-26, 2010*, Sung Y. Shin, Sascha Ossowski, Michael Schumacher, Mathew J. Palakal, and Chih-Cheng Hung (Eds.). ACM, 867–871.

Jon M. Kleinberg. 1999. Hubs, authorities, and communities. *Comput. Surveys* 31, 4es (1999), 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 (1997), 211–240. Issue 2.

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, Paris, France, June 28 - July 1, 2009*, John F. Elder IV, Françoise Fogelman-Soulié, Peter A. Flach, and Mohammed Javeed Zaki (Eds.). ACM, 467–476.

Baichuan Li and Irwin King. 2010. Routing questions to appropriate answerers in community question answering services, See Huang et al. [2010], 1585–1588.

Baichuan Li, Irwin King, and Michael R. Lyu. 2011. Question routing in community question answering: putting category in its place, See Macdonald et al. [2011], 2041–2044. DOI:http://dx.doi.org/10.1145/2063576.2063885

Jing Liu, Young-In Song, and Chin-Yew Lin. 2011. Competition-based user expertise score estimation. In *Proceeding of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2011, Beijing, China, July 25-29, 2011*, Wei-Ying Ma, Jian-Yun Nie, Ricardo A. Baeza-Yates, Tat-Seng Chua, and W. Bruce Croft (Eds.). ACM, 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, 11th International Conference, WAIM 2010, Jiuzhaigou, China, July 15-17, 2010. Proceedings (Lecture Notes in Computer Science)*, Lei Chen, Changjie Tang, Jun Yang, and Yunjun Gao (Eds.), 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, Bremen, Germany, October 31 - November 5, 2005*, Otthein Herzog, Hans-Jörg Schek, Norbert Fuhr, Abdur Chowdhury, and Wilfried Teiken (Eds.). ACM, 315–316.

Craig Macdonald, Iadh Ounis, and Ian Ruthven (Eds.). 2011. *Proceedings of the 20th ACM Conference on Information and Knowledge Management, CIKM 2011, Glasgow, United Kingdom, October 24-28, 2011*. ACM.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013a. Distributed Representations of Words and Phrases and their Compositionality. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States.*, Christopher J. C. Burges, Léon Bottou, Zoubin Ghahramani, and Kilian Q. Weinberger (Eds.). 3111–3119.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013b. Linguistic Regularities in Continuous Space Word Representations. In *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 9-14, 2013, Westin Peachtree Plaza Hotel, Atlanta, Georgia, USA*, Lucy Vanderwende, Hal Daumé III, and Katrin Kirchhoff (Eds.). The Association for Computational Linguistics, 746–751.

Alain Mille, Fabien L. Gandon, Jacques Misselis, Michael Rabinovich, and Steffen Staab (Eds.). 2012. *Proceedings of the 21st World Wide Web Conference, WWW 2012, Lyon, France, April 16-20, 2012 (Companion Volume)*. ACM.

Jeff Mitchell and Mirella Lapata. 2010. Composition in Distributional Models of Semantics. *Cognitive Science* 34, 8 (2010), 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, held at ICML 2010, Haifa, Israel, June 25, 2010 (JMLR Proceedings)*, Olivier Chapelle, Yi Chang, and Tie-Yan Liu (Eds.), Vol. 14. JMLR.org, 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*, Rob Gaizauskas, Mark Greenwood, and Mark Hepple (Eds.). 23–30.

Meredith Ringel Morris, Jaime Teevan, and Katrina Panovich. 2010. A Comparison of Information Seeking Using Search Engines and Social Networks, See Cohen and Gosling [2010].

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 2009, Boston, MA, USA, July 19-23, 2009*, James Allan, Javed A. Aslam, Mark Sanderson, ChengXiang Zhai, and Justin Zobel (Eds.). ACM, 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. Previous number = SIDL-WP-1999-0120.

Katrina Panovich, Rob Miller, and David R. Karger. 2012. Tie strength in question & answer on social network sites. In *CSCW '12 Computer Supported Cooperative Work, Seattle, WA, USA, February 11-15, 2012*, Steven E. Poltrock, Carla Simone, Jonathan Grudin, Gloria Mark, and John Riedl (Eds.). ACM, 1057–1066.

Fatemeh Riahi, Zainab Zolaktaf, M. Mahdi Shafiei, and Evangelos E. Milios. 2012. Finding expert users in community question answering, See Mille et al. [2012], 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 *ACL 2007, Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics, June 23-30, 2007, Prague, Czech Republic*, John A. Carroll, Antal van den Bosch, and Annie Zaenen (Eds.). The Association for Computational Linguistics.

Gerard M. Salton, A. Wong, and C. S. Yang. 1975. A vector space model for automatic indexing. *Commununications of ACM* 18 (November 1975), 613–620. Issue 11.

Hinrich Schütze. 1998. Automatic Word Sense Discrimination. *Computational Linguistics* 24, 1 (1998), 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, See con [2008].

Pavel Serdyukov and Djoerd Hiemstra. 2008. Modeling Documents as Mixtures of Persons for Expert Finding. In *Advances in Information Retrieval , 30th European Conference on IR Research, ECIR 2008, Glasgow, UK, March 30-April 3, 2008. Proceedings (Lecture Notes in Computer Science)*, Craig Macdonald, Iadh Ounis, Vassilis Plachouras, Ian Ruthven, and Ryen W. White (Eds.), 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 2008, Napa Valley, California, USA, October 26-30, 2008*, James G. Shanahan, Sihem Amer-Yahia, Ioana Manolescu, Yi Zhang, David A. Evans, Aleksander Kolcz, Key-Sun Choi, and Abdur Chowdhury (Eds.). ACM, 1133–1142.

Aliaksei Severyn and Alessandro Moschitti. 2012. Structural relationships for large-scale learning of answer re-ranking, See Hersh et al. [2012], 741–750.

Mário J. Silva, Alberto H. F. Laender, Ricardo A. Baeza-Yates, Deborah L. McGuinness, Bjørn Olstad, Øystein Haug Olsen, and André O. Falcão (Eds.). 2007. *Proceedings of the Sixteenth ACM Conference on Information and Knowledge Management, CIKM 2007, Lisbon, Portugal, November 6-10, 2007*. ACM.

Elena Smirnova and Krisztian Balog. 2011. A User-Oriented Model for Expert Finding. In *Advances in Information Retrieval - 33rd European Conference on IR Research, ECIR 2011, Dublin, Ireland, April 18-21, 2011. Proceedings (Lecture Notes in Computer Science)*, Paul D. Clough, Colum Foley, Cathal Gurrin, Gareth J. F. Jones, Wessel Kraaij, Hyowon Lee, and Vanessa Murdock (Eds.), 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 (1990), 159–216.

Mark D. Smucker, James Allan, and Ben Carterette. 2007. A comparison of statistical significance tests for information retrieval evaluation, See Silva et al. [2007], 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 (2011), 351–383.

Joseph P. Turian, Lev-Arie Ratinov, and Yoshua Bengio. 2010. Word Representations: A Simple and General Method for Semi-Supervised Learning. In *ACL 2010, Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, July 11-16, 2010, Uppsala, Sweden*, Jan Hajic, Sandra Carberry, and Stephen Clark (Eds.). The Association for Computer Linguistics, 384–394.

Peter D. Turney. 2006. Similarity of Semantic Relations. *Computational Linguistics* 32, 3 (2006), 379–416.

Peter D. Turney and Michael L. Littman. 2005. Corpus-based Learning of Analogies and Semantic Relations. *Machine Learning* 60, 1-3 (2005), 251–278.

Peter D. Turney and Patrick Pantel. 2010. From Frequency to Meaning: Vector Space Models of Semantics. *Journal of Artificial Intelligence Research (JAIR)* 37 (2010), 141–188.

Suzan Verberne, Lou Boves, Nelleke Oostdijk, and Peter-Arno Coppen. 2008. Using Syntactic Information for Improving Why-Question Answering. In *COLING 2008, 22nd International Conference on Computational Linguistics, Proceedings of the Conference, 18-22 August 2008, Manchester, UK*, Donia Scott and Hans Uszkoreit (Eds.). 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 (2010), 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 (2011), 107–132.

Dominic Widdows and Kathleen Ferraro. 2008. Semantic Vectors: a Scalable Open Source Package and Online Technology Management Application, See con [2008].

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

Michael B.W. Wolfe, M.E. Schreiner, Bob Rehder, Darrell Laham, Peter W. Foltz, Walter Kintsch, and Thomas K. Landauer. 1998. Learning from text: Matching readers and texts by latent semantic analysis. *Discourse Processes* 25, 2-3 (1998), 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 2013, 4-9 August 2013, Sofia, Bulgaria, Volume 1: Long Papers*. The Association for Computer Linguistics, 1744–1753.

ChengXiang Zhai and John D. Lafferty. 2001. A Study of Smoothing Methods for Language Models Applied to Ad Hoc Information Retrieval. In *SIGIR 2001: Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, September 9-13, 2001, New Orleans, Louisiana, USA*, W. Bruce Croft, David J. Harper, Donald H. Kraft, and Justin Zobel (Eds.). ACM, 334–342.

Jun Zhang, Mark S. Ackerman, and Lada Adamic. 2007a. CommunityNetSimulator: Using Simulations to Study Online Community Networks. In *Communities and Technologies 2007*, Charles Steinfield, Brian T. Pentland, Mark Ackerman, and Noshir Contractor (Eds.). Springer London, 295–321.

Jing Zhang, Jie Tang, and Juan-Zi Li. 2007b. Expert Finding in a Social Network. In *Advances in Databases: Concepts, Systems and Applications, 12th International Conference on Database Systems for Advanced Applications, DASFAA 2007, Bangkok, Thailand, April 9-12, 2007, Proceedings*, Kotagiri Ramamohanarao, P. Radha Krishna, Mukesh K. Mohania, and Ekawit Nantajeewarawat (Eds.). 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 *IJCAI 2013, Proceedings of the 23rd International Joint Conference on Artificial Intelligence, Beijing, China, August 3-9, 2013*, Francesca Rossi (Ed.). IJCAI/AAAI.

Tom Chao Zhou, Michael R. Lyu, and Irwin King. 2012. A classification-based approach to question routing in community question answering, See Mille et al. [2012], 783–790.

Hengshu Zhu, Huanhuan Cao, Hui Xiong, Enhong Chen, and Jilei Tian. 2011. Towards expert finding by leveraging relevant categories in authority ranking, See Macdonald et al. [2011], 2221–2224.