Chapter 9
Future Directions of Query Understanding

David Carmel, Amazon Research
Yi Chang, Jilin University
Hongbo Deng, Alibaba Inc.
Jian-Yun Nie, University of Montreal

Abstract Query understanding is to bridge the gap and establish the communication channel between the searcher and the search engine. This raises a huge challenge for going beyond traditional keyword query. One bigger challenge is to interact with search engines in a more natural way, such as in the form of natural language, question answering or conversation, and so on. In addition, how to employ additional information, such as knowledge graph and cross-language, to assist query understanding, becomes more and more important. Moreover, there are a lot of related questions and settings in query understanding that have not been yet fully explored. We will review some of them in this chapter and hope that researchers interested in query understanding will try to answer these challenging and exciting research questions.

9.1 Personalized Query Understanding

Query understanding is essentially limited if the user's personal perspective is not taken into consideration. Different people specify the same information need in different manners and the relevance of an item to the query is varied according to the user's private interests, prior knowledge, and the current context of the search session.

Personalized query understanding (PQU) is the initial process of personalized search which analyzes the user query according to the user's specific needs, personal knowledge, and the context she is currently involved with. Search personalization has been extensively studied by the IR community (e.g., [62, 34]) and was invoked,

to some extent, by all commercial search engines[1]. In this section we briefly discuss our own anticipation how PQU is expected to emerge in the coming future.

Search Personalization can be done at certain levels of granularity. The most basic one is session analysis, where the user's query is analyzed with respect to the previous queries submitted during the user's current search session, and the responses as reflected by the user feedback on the search results [18]. For example, a user searching for "parking" while her previous query was "Golden Bridge, SF", should only be exposed to parking lots in the Golden bridge area. Other parking lots are unlikely to be relevant in this specific session. Similarly, previous search results and the corresponding user feedback, should also be taken into account while analyzing the current query, e.g., by downgrading results that have already been clicked (or ignored) previously during the current search session [71].

While current instrumentation tools for session analysis are mostly based on the user on-line feedback, as reflected through her clicks, mouse tracking, and her abandonment rate [15], much better instrumentation tools for measuring user engagement are expected to emerge, such as eye tracking, face expression analysis, sentiment analysis, and many more. Such tools would let us better analyze the user satisfaction (or dissatisfaction) with the search results, thus letting us tuning our search engine for better understanding and serving our users.

The long history of the user interaction with the search engine also provides important clues about the user general interests [6]. Analyzing the current query in the context of the user's search history, e.g., by topic modeling, can assist in understanding the user general topics of interests, thus assisting us in query disambiguation and classification [29]. Current search personalization approaches are mostly based on analyzing previous queries and previously visited Web pages. It is very likely that in the close future many other types of user feedback, on any digital device, could be tracked, aggregated, and be used for better modeling the user interests [79]. For example, the list of applications that we use on our smart-phones on a daily basis are extremely effective in identifying our interests and goals [4]. Another example is the user activity on social media sites where the user posts, comments, and shares, provide valuable data about her areas of interest. The user own social network can be furthered analyzed for better understanding the topics and issues that are relevant to the user in the context of her community [12]. Analyzing such rich types of data sources will enrich our understanding of the users goals and preferences and will let us to better serve their information needs.

While PQU is going to emerge significantly in the typical search scenario, it is also critical and essential for personal digital assistants like Siri[2], Cortana[3], and Alexa[4]. These agents are expected to answer our questions, make orders for us in on-line shopping sites, recommend relevant content, assist us in organizing our travels, etc. Such assistants require advanced personalization capabilities in order to keep track of

[1] For example, https://googleblog.blogspot.co.il/2009/12/personalized-search-for-everyone.html

[2] https://www.apple.com/ios/siri/

[3] https://www.microsoft.com/en-us/windows/cortana

[4] https://www.alexa.com/

our knowledge, preferences, and the context we are currently involved with, in order to serve us optimally. For example, when ordering coffee from our favorite coffee shop, my personal assistance is expected to be aware of that I drink my coffee with cream, no sugar, and very hot, while my wife drinks it black and weak. When asking for recommendation for a birthday present for Jenny, my assistant should know that Jenny is my fifth years old daughter. When asking our agent to order shampoo for our family, it should be aware of the types of shampoo favored by all family members, our favorite suppliers, as well as all other relevant details.

The main tool for capturing personalized data is a personalized knowledge graph (PKG) which will encapsulate all related entities of the user such as family members, friends, neighbors, contacts, as well as preferences, biases, and interests. The PKG will complement the general knowledge graph (KG) that is already being widely used by search engines for providing up-to-date information about popular entities such as politicians, celebrities, organizations, products, locations, etc. The PKG will be focused on entities strongly relevant to the user. Our personal social network, locations (home, work, frequent visiting sites), medicines, dietary ingredients, media preferred entities, should all be represented in our PKG. The PKG will be used by the assistant agents to personalize the interaction with the user. Each query will be analyzed by considering the personal entities in this graph, in addition to the entities extracted from the general KG, and their relationships with the user.

To summarize, we can safely anticipate that query understanding will become much more personalized in the coming future for supporting deep personalized search experience, provided through general-purpose search engines as well as through personal digital assistants.

9.2 Natural Language Question Understanding

Another popular trend in the IR domain is moving from keyword queries to natural language questions. Current mobile devices enable users to input spoken language queries into their search applications, taking advantage of recent developments in speech recognition technology that exceeds human performance in spoken language understanding [65]. Spoken queries are typically much longer and are usually pronounced as natural language questions, rather the standard keyword queries that we are used to issue in current Web search services [28].

In contrast to short keyword queries, long queries can benefit from Natural Language Processing (NLP) methods. While NLP analysis for short queries typically fails to bring significant improvement over shallow statistical-based methods, they were found useful for long queries where syntactic analysis such as part-of-speech tagging and dependency parsing complement standard statistical term weighting methods [14].

Serving natural language questions strongly corresponds with the traditional question answering task which has been mostly focused on answering factoid questions [40]. The standard flow of question answering process begins with question anal-

ysis for identifying the lexical answer type, i.e., the category type of the answer expected for that question (e.g. country, capital city, date, distance). Then, passages are identified in a given knowledge-base that are likely to contain an answer to the question. Candidate answers are then extracted from the top retrieved passages and are judged and scored according to many criteria. The top scored candidate is then selected for the final answer. A typical judge, for example, will filter out candidates not belonging to the question's lexical category type identified during the question analysis phase. This paradigm was successfully demonstrated by IBM Watson which was able to outperform human trivia experts in the game of Jeopardy [24]. However, even the extremely complicated Jeopardy questions are limited to factoid questions only. More complex needs such as why questions, opinion and advice seeking questions, puzzles, and many other types, are still an open challenge and deserve further research for understanding the actual information need behind them.

Another emerging direction for question understanding is the identification of Web queries having a question intent which constitute about 10% of the issued queries [80]. Such queries, even formulated as keyword queries, seek for a direct and detailed answer rather than a list of search results. Current Web search engines usually handle such queries by developing a specific tool for any specific question type. Weather-based queries are served by the Weather agent while stock-based queries are handled by the Finance agent. The same approach is taken for handling named-entity queries where the entity's relevant information, extracted from the general-purpose knowledge graph, is directly displayed on the SERP enriching the standard Web search results.

Furthermore, a new trend emerges recently of handling factoid questions by existing question answering techniques. This approach is immature yet and in its infant stages but we can expect significant progress in the future. Complementary, any question-intent queries can be served by searching over an archive of community question answering sites, looking for similar questions that have already been manually answered by humans. This approach was dominant among participants in the TREC's Live-QA track [1] where participants were challenged to answer real human questions in real time (in less than one minute). Real human questions submitted on the Yahoo Answers site, were submitted to participant systems during the contest and were answered automatically and immediately by the participant systems. Most participants searched for the answer over a given archive of question-answer pairs to provide the most appropriate human answer for similar questions. Many approaches were examined for measuring the relevance of question-answer pairs to the given question. One interesting technology presented in the track was a combination of automatic search with human judgment; a list of candidate answers were retrieved by the search component and then were judged in real-time by crowdsourcing humans [66]. One of the conclusion of the Live-QA challenge was that while previously-answered questions can be useful to answer popular questions, advanced answer generation techniques should be considered in order to answer, with high quality, long-tailed questions.

To conclude, the trend of moving from keyword queries to natural language questions enables users to better express their needs, and to easily provide their

questions through much more diverse and highly accessible input devices. However, these complicated questions open many new challenges in question analysis and question understanding, and require the development of advanced techniques that should be further explored.

9.3 Dialog Query Conversational Query Understanding

The current search engines mainly focus on one-shot search: the search results are basically determined by the current query the user has formulated. Few attempts have been made to engage a conversation with the user to better understand the search intent of the user. The burden is on the user who has to learn to adapt to the search systems: when a query was not successful, the user has to modify it based on an analysis of the previous search results. Such modifications can be repeated several times before the user can find the desired documents. Even though, it is not rare to see frustrated users who fail to retrieve desired documents and to understand why their queries have not been successful. The interface of search engines is not user friendly and does not provide much help to the user to formulate better queries

Looking back into the history, IR was imagined as an intermediary between the library system and the user – a role that was played typically by a librarian. To understand what the user was looking for, the librarian usually held a conversation (negotiation) with the user to understand the information need of the user and to generate a good search query to be submitted to a library system [78]. Even though we do not think about using a human intermediary for search nowadays, or have the luxury to do it, the existence of a human intermediary provided at least several advantages compared to the current interface:

- She/he knows better the useful search terms to use than most users, being familiar with the data collections;
- She/he knows better databases to search (when there are multiple search systems);
- She/he understands the search intent of the user.

These advantages are precious for users who are not familiar with the search engine, the documents indexed, or the searched topics. A conversational intermediary can play a similar role as human librarian to help the user. Some typical cases where the conversational assistance can be helpful are as follows:

- The user's initial query is ambiguous: either ambiguous terms are used, or the whole query may lead to very different types of documents. If ambiguity is detected, a clarification question can be asked to the user [2].
- The query is under-specified: The query may be too general or too vague, leading to too many search results. It may be useful to ask the user to provide more details about the searched topics. For example, some choices can be offered to the user based on the distribution of the corresponding topics [72].
- The formulated query does not contain the best search terms. When formulating a query, a user may not have the experience to choose the best search terms. In this

case, the conversational assistance can suggest better terms or a better formulation of the query.
- A search topic may be strongly related to other topics, which could be of interest to the user. For example, it may be useful to the user to also learn about the background information when searching about an event, or to learn about its next evolution [8]. The conversational assistance can take a proactive role to suggest related topics to users.

While conceptually, the above assistance can be useful, it has to be implemented correctly. A bad assistance tool can easily become annoying. To implement effective conversational assistance to understand search intents, we are faced with the following technical challenges:

- How to detect is a conversational assistance is needed?
- How to determine the best action? Should the system ask a clarification question? provide some results and see how the user interact with them? or suggest alternative queries/topics?
- How to generate a natural and relevant reply or question? This aspect is particularly challenging for the current conversation technology, which is able to generate replies in task-oriented conversation in limited domains with predefined knowledge structure, but has difficulty to do it in open-domain conversation [2]. A key issue to investigate is whether it is possible to develop some general conversation patterns for general search tasks. For example, when a query ambiguity is detected, a clarification question such as "do you mean X or Y by [original query]?" can be generated. To suggest alternative queries, the system can suggest "try the query [suggested query] that has been successful for other users", or "your search topic is related to [suggested query]".
- How to judge the success of a conversational query understanding process? The goal of new interaction methods, including conversational query understanding, is to help the users to do more effective search. When the user is involved in the loop, the current evaluation methodology becomes insufficient. Some attempts have been made to evaluate the search process in which the user participate [37], but there is still no general consensus on the appropriate methodology for conversational IR.
- Finally, we also have to think about the possible forms of a conversational assistance. Dialog in natural language (either in speech or in text) is the first form of conversation we can think about. Should we limit conversational assistance to this narrow form, or should we give conversation a wider meaning, to include other forms of interactions such as providing choices to the user, let the user click on some results? [49]

In summary, conversational query understanding and assistance will likely change the face of search engines in the future, but many underlying problems remain to be explored and solved to make it effective in practice.

9.4 Medical Query Understanding

Medical IR is an important application area. People often use search engine to locate relevant information in addition to consulting physicians. However, the current search engines are limited in providing appropriate search tools in this specific area. In most cases, users are left with a search engine constructed with the general technology, even though the documents in the database may be in the medical domain. A good understanding of medical queries is particularly important because most users are not familiar with the specialized concepts used in the medical documents. This situation also makes the understanding very challenging. Some of the main difficulties are as follows:

- Vocabulary mismatch: End users may not know the exact specialized term of a medical concept. Even though some lexical resources have been constructed, trying to bridge the vocabulary gap between specialized documents and non-specialized end users [91], they are far from enough to solve the problem. The problem of vocabulary mismatch is not limited to the level of words or terms, it can be at a more global level. For example, a user may use several sentences to explain a health condition, which could be described by a specialized term.
- Concept mapping: A strongly related problem is to recognize correctly the concepts described in a text (a document or a query). This is a key step for correct query understanding. Concept mapping in medical domain has attracted a large amount of research work. Most approaches leverage the existing lexical resources (e.g. UMLS Metathesaurus[5]) and make use of syntactic rules, variations on word forms, and statistics to determine what concepts a sequence of words can correspond to. MetaMap[6] [3] is considered to be one of the best tools in this area. However, its accuracy on query analysis was estimated at only about 70% [21, 67], making it difficult to rely on for document matching.
- Exploring more resources to learn concept mapping: The existing research on concept mapping has been limited to lexical mapping an observed sequence of words to the possible expressions of a concept in a lexical resource. The recent development on deep learning offers us a great opportunity to match a piece of text with a concept in a latent representation space: Both concepts and words/sentences could be mapped into the same representation space, allowing them to be directly compared. While some preliminary studies in this direction has been done [47, 48] showing promising results, more investigations are required to fully explore the potential of this approach.
- In addition to search queries, users tend to ask more complex questions. In forums of discussions where users can ask questions to physicians or other peer users, it is common to see long questions with a description about the patient and the problem, and asking for advice. While we do not see, in the current stage, that human replies can be completely replaced by automatic replies, it is

[5] https://www.nlm.nih.gov/research/umls/knowledge_sources/metathesaurus/

[6] https://metamap.nlm.nih.gov/

useful to process such long and complex questions to help users locate the most useful documents or pieces of information. We are then faced with the problem of understanding complex medical questions, which is not limited to merely identifying the key concepts involved, but also relate them so as to construct a complete picture (graph) about the question. For example, we should not only recognize that the user's question involves the concept "pneumonia", but also that "pneumonia" happened to the patient one month ago rather than now, and the patient is a 50 years-old adult. This fine-grained analysis is crucial in this area.

In addition to query understanding, documents should also be understood in a similar way. Finally, new matching processes are required to compare complex query and document representations. All these problems require more research work.

9.5 Cross Language Query Understanding and Translation

In the majority of cases, users are interested in searching documents in the same language as the query. However, this situation does not mean that there is no need for searching documents in other languages. Cross-language and multilingual search is needed in several typical cases [25]. For example, the topic may not be well covered in the language of the query, but is well covered in another language; or the search needs to be complete (recall-oriented search such as patent retrieval) in all languages. In these typical cases, a search query has to be translated into one or several other languages. Query translation is a challenging task.

A general machine translation can do a good job for translating most queries: when there is no ambiguity and when terms in a query have a clear translation in another language. However, we are often faced with the translation ambiguity problem, especially for short queries which provide limited context information. The existing approaches have explored the utilization of the following information to select good translation terms:

- Translation probability
- How common a term is used in the target language

In addition to the problem of translation, query translation also plays the role of selection of good search terms: When several translation alternatives exist, it may be better to select the one that is more discriminative, or to combine all of them. The inclusion of multiple translation terms in query translation has naturally produced a desired effect of query expansion [25, 82].

Cross language query understanding is not limited to translation only. The search behaviors in different language communities could be different. For example, while people in North America are more concerned with water and soil pollution, people in China can be more concerned with air pollution. So a search on "pollution" in different language communities may lead to different results. Cross language query understanding could be extended to the understanding of search intents in different

language communities, and when possible, making the required adaptation. This has been found very useful in some existing work [25].

The further development on cross language query understanding will certainly benefit from the development of deep learning approaches. Indeed, if both the query and the document can be mapped into a common representation space, whatever their language is, then the translation problem does not exist anymore. Such interlingua representation has been investigated in recent MT studies [23], which assumes that different languages share a common representation space, in addition to a private space specific to each language. However, much more investigations are required to make the approach effective in practice.

9.6 Temporal Dynamics of Queries

The World Wide Web is highly dynamic and is constantly evolving: as a large number of new web pages are created or updated every second, information on those old web pages are outdated quickly. At the same time, web search is strongly influenced by time: some queries occasionally spike in popularity, some queries periodically spike, and others remain relatively constant. In order to help search engine users to find the latest updated information, it is foremost to detect those time-sensitive queries and understand their temporal dynamics, which benefits not only search ranking [22], but also query auto-completion [70, 11].

Given a query, we count its frequency during a pre-defined time interval, and generate a time-series about this query. In order to model the temporal shapes, power Law distribution is proposed as the function to model burst time-series [17], and recently Hawkes process is leveraged to model temporal bursts with multiple spikes [64]. Another useful approach is to model occurrence of spikes using infinite-state automation approach [39]. Yet, the method uses spike locations as input, and thus it is not possible to directly apply the infinite-state automation approach for raw time-series data.

In addition, temporal information help us to group topics together. Once a sudden spike appears on the extracted time-series, most likely, many of users are searching the same topic or the same event, which indicates a strong relationship between content information and temporal information. Therefore, it is necessary for us to combine temporal information with content information into the same framework, yet it is a very challenging and difficult task, as temporal shapes and textual content are heterogeneous. To combine temporal modeling with content analysis is not brand new, and there are a few excellent work [35, 42]. However, these existing work either assumes topical distribution changes smoothly or just models temporal information as a sequence of bursts, which could not explicitly model the temporal shapes with the sudden spikes.

Furthermore, temporal dynamics of queries can be leveraged for prediction. Since whether a query is triggered by an event can be successfully predicted [64], it is possible to improved query auto-completion by leveraging terms related to the

triggered event, or to enhance search result ranking via boosting documents related to the triggered event, which are promising research ideas. Yet, how to handle prediction with intent shifting or triggered by multiple events are still unsolved open challenges.

9.7 Deep Learning for Query Understanding

With the success of deep learning in many research areas, Information Retrieval (IR) community has started to explore deep learning based techniques to various query understanding problems. The key features of deep learning are representation learning and end-to-end training. We begin by introducing different neural approaches to learn vector representations of queries. We then review some shallow and deep neural methods that employ pre-trained word embeddings as well as learn the end-to-end query understanding task such as query expansion, spelling correction, query classification, and so on.

Vector representations are fundamental to both information retrieval and deep learning. Different vector representations exhibit different level of generalization and could derive different level of similarity. In traditional IR, query and document are represented as bag of words, and many approaches rely on exact term matching between the query and the document text. To be able to perform soft term matching between semantically similar words, a number of studies have focused in particular on the use of word embeddings generated using shallow or deep NNs. For example, the term 'hotel' and 'motel' are two separate words which cannot match each other with bag of words, while ideally they could share a large similarity using word embeddings. Word embedding, also known as distributed representation of words, refers to a set of machine learning algorithms that learn high-dimensional real-valued dense vector representation $\mathbf{w} \in R^d$ for each vocabulary term w, where d denotes the embedding dimensionality. Word2vec [52] and GloVe [60] are two well-known word embedding algorithms that learn embedding vectors in an unsupervised learning. The underlying idea is that the words that often appear in surrounding contexts are similar to each other. Such word embeddings can be used to capture a certain type of topical similarity, such as 'hotel' to be similar to 'motel', and 'wife' to be similar to 'husband'. It is worth noting that learning different word embeddings can capture different types of similarities, which may not be appropriate for a certain retrieval scenario.

A better alternative is to learn embeddings as a set of parameters in an end-to-end neural network model for a specific IR task [85, 19, 90]. The word embeddings can be aggregated in different ways for estimating query embedding vectors, and using the average word embeddings is quite popular [44, 81, 56]. In [88], a theoretical framework has been proposed with different implementations for estimating query embedding vectors based on individual word embeddings, which shows average word embeddings is a special case. In addition, Dehghani et al. [19] proposed to represent query as a weighted sum of word embeddings by learning the global weight for each

term in the vocabulary set. Training word embedding vectors based on additional data, like query logs and click-through data, was also studied in [32, 73, 26]. Recently, Grbovic et al. [27] used query embeddings to include session-based information for sponsor search. Estimating accurate query embedding vectors can improve the performance of many of the embedding-based methods that need to compute query vectors. It should be noted that in a realistic case, many tail and rare queries are not available during the training time of embedding vectors, which makes direct training of query embedding vectors problematic. How to learn the embeddings for tail and rare queries is still a very challenge task.

There are many existing works [73, 20, 43, 89, 5] that attempt to leverage word embeddings for query expansion. One straightforward method [43, 89, 5] is to employ the pre-trained term embeddings to select terms that are similar to the query as a whole or its constituent terms, and then the selected terms are used to expand the query in a unigram language model framework. For example, Zamani and Croft [89] presented a set of embedding-based query language models using the query expansion and pseudo-relevance feedback techniques that benefit from the word embedding vectors. Diaz et al. [20] proposed to train word embeddings on topically-constrained corpora, instead of large topically-unconstrained corpora. These locally-trained embedding vectors were shown to perform well for the query expansion task. Zheng and Callan [94] proposed a supervised embedding-based term re-weighting technique applied to the language modeling and BM25 retrieval models.

Convolutional neural networks (CNNs) and recurrent neural networks (RNNs) are two most common architectures, where CNNs were originally developed for image classification [33, 41] and RNNs have been successfully used in natural language processing [31, 51]. Recently, a number of deep neural networks with deep architectures have been applied to some specific query understanding approaches. For example, CNN and RNNs have emerged as top performing architectures in query classification and query intent detection [38, 36, 92, 30, 69]. Park and Chiba [59] proposed a neural language model with recurrent layers for query auto-completion task. Another special type of neural network architecture is Siamese networks. A Siamese networks consists of two identical neural networks, each taking one of the two inputs, such as the query and the document. The last layers of the two networks are then fed to a contrastive loss function, which calculates the similarity between the two inputs. The Deep Semantic Similarity Model (DSSM) [32] is on such architecture that trains on query and document title pairs and learns the similarity between them. Convolutional-DSSM (CDSSM) [68] employs more sophisticated architectures involving convolutional layers. Mitra and Craswell [54] trained the same CDSSM architecture using query prefix-suffix pairs and leveraged the model to suggest query completions for rare query prefixes. Obviously, an appropriate network architecture makes big difference for end-to-end training, but it can be difficult to determine when to use which kind of network architectures. For a given query understanding task, predicting in advance which will work best is usually impossible, and how to design an appropriate network architecture remains an open question.

Some deep learning methods operate at the character-level or character n-gram [32, 68, 55]. For instance, the deep learning method for spelling correction is usually sequence-to-sequence models. A sequence-to-sequence model [76] consists of an encoder and decoder. The encoder converts a sequence of characters or tokens, into a single vector, while the decoder begins with this vector, and it keeps generating characters or tokens until it generates a special stop symbol. Note that the lengths of the source and target sequences don't need to be the same. Both the encoding and decoding are done using RNNs. Xie et al. [84] presented an encoder-decoder RNN with an attention mechanism by operating at the character level. Sordoni et al. [74] formulated a hierarchical recurrent encoder-decoder architecture and used it to produce query suggestions which takes account for sequences of previous queries of arbitrary lengths as context. Another advanced query reformulation system proposed by Nogueira and Cho [57] is to train neural network with reinforcement learning. The actions correspond to selecting terms to build a reformulated query, and the reward is the document recall.

9.8 Semantic Understanding and Matching for Search Queries

Semantic matching is one of the most difficult challenges especially for tail queries [45]: query document mismatch occurs when the queries and documents use different terms to describe the same concept. For instance, for the query "how much is tesla", relevant documents may contain the term "price" rather than "how much", so the widely used bag-of-words approach is insufficient to solve this challenge [87].

The basic idea of semantic matching is to project a query or a document directly or indirectly into its semantic space separately, and then match the query and the document on their semantic spaces. The traditional semantic matching approach can be grouped into the following categories:

Semantic Matching with Machine Translation Model. Its basic idea is to leverage machine translation models to deal with query and document mismatching, which is a supervised learning method. In particular, queries are considered as the source language, while the clicked documents derived from click-through data are considered as the target language, then search can be formulated as a statistical machine translation problem [7], which query q is translated into document d with the largest conditional probability $P(d|q)$.

Semantic Matching with Topic Model. It is well known that queries and documents with the same topic are more likely to be considered as relevant, as their semantic are consistent at the topic level. The basic idea of this approach is to use topic models, such as LDA or PLSI, to obtain the topics of each query or each document, then leverage topic matching techniques to deal with query document mismatching, which could successfully improve search relevance [86]. Generally speaking, this approach belongs to unsupervised learning.

Semantic Matching with Latent Space Model. In this approach, queries and documents are trained to mapped into the same latent space, and the semantic matching function is defined as the inner product between the projection of the query and the projection of the document in the latent space, while each dimension of the latent space does not necessarily have its corresponding semantic meaning [32]. Generally speaking, this approach is a supervised learning approach.

Semantic Matching with Deep Learning Model. Recently work on semantic matching mainly based on deep learning algorithms [58], which can automatically learn relations among words from vast amount of search log data, and make a fully use of information from phrase patterns and text hierarchical structures, and experimental results usually show a better performance.

In fact, these different approaches of semantic matching are complementary, and how to effectively combine them into one generic framework is an open question. In addition, how to handle the semantic matching when queries are too short is still a challenging problem, since deep learning based text matching approach works well when the queries are relatively lengthy. Furthermore, how to handle multimodal semantic matching is another challenging problem, such as the semantic matching between a text query and an image, or between an image query and a text document, which is critical for image search and video search in commercial search engines.

9.9 Query Understanding with Knowledge Graph

Knowledge bases, better known as knowledge graphs, such as Wikipedia, DBpedia, Freebase [10] and Yago [75], have emerged in recent years. Most of them are encyclopedic knowledge bases, containing entities and facts such as Barack Obama's birthday and birthplace. The knowledge graphs have been utilized for enhancing query understanding in an entity-aware way for the rich facts organized around entities. For example, Google took the first step understanding and answering queries with the knowledge graph in 2012, and they started by providing information on individual entities like "Barack Obama" or "Brad Pitt". Recently, search engines become a little bit smarter, and could answer simple questions about those entities, such as "How old is Barack Obama?" or "Who are the author of Harry Potter?". All of these works rely on query understanding with knowledge graph. There are a few challenges as listed below:

First, a widely accepted way to use knowledge graph is to annotate the entities in the query and link them to a knowledge base, also known as entity linking. TagME [61] is a very early work on entity linking in queries. It generates candidates by searching Wikipedia page titles, anchors and redirects, then exploit the structure of the Wikipedia graph for disambiguation. Entity linking in queries is also viewed as the problem of finding multiple query interpretations [13], usually with three phrases: fetching, candidate-entity generation and pruning. One challenge is that the queries are usually very short and contain insufficient information, thus it becomes

very important to leverage additional information, such as Wikipedia [16, 77], query log and search results [9].

Second, quite a few non-entity words are barely included in knowledge graph, and knowledge about how words interact with each other in a language (instead of encyclopedia knowledge) plays an important role in query understanding. As we discussed above, the encyclopedia knowledge base contains entities and facts, while the other type of knowledge base is mainly about common sense or linguistic knowledge among terms, such as KnowItAll [50], NELL [53] and Probase [46]. For non-entity words, recently there appears a tendency to mine a variety of relations among terms and map them to related concepts [83] or intent topics [93], then propagate the enriched features in a graph consisting of concepts or intent topics using an unsupervised algorithm. How to effectively extract knowledge of non-entity words and represent them in a unified knowledge graph remains a challenging task for query understanding.

Third, with the extensive knowledge graph, structured query understanding is a critical component to improve the relevance of search engines. For example, identifying attributes in a query for e-commerce platforms could significantly improve the performance in connecting users to relevant items. In many cases, the queries might have multiple attributes, and some of them will be in conflict with each other. Leveraging the e-commerce catalog [63] as an additional knowledge base to supplement the textual information can help to resolve conflicting query attributes. Similarly, additional domain-specific knowledge graph will be very valuable for structured query understanding in other domains, such as healthcare.

As discussed above, the knowledge graph makes it possible to break down a query to understand the semantics of each piece and get the intent behind the entire query. Moreover, that makes it reliable to traverse the knowledge graph to find the right facts and compose a useful answer for a given query.

References

[1] Eugene Agichtein, David Carmel, Dan Pelleg, Yuval Pinter, and Donna Harman. Overview of the TREC 2015 liveqa track. In *Proceedings of The Twenty-Fourth Text REtrieval Conference*, volume 500-319, 2015.

[2] Mohammad Aliannejadi, Hamed Zamani, Fabio Crestani, and W. Bruce Croft. Asking clarifying questions in open-domain information-seeking conversations. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 475–484, 2019.

[3] Alan R. Aronson. Effective mapping of biomedical text to the UMLS metathesaurus: the metamap program. In *Proceedings of the American Medical Informatics Association Annual Symposium*, pages 17–21, 2001.

[4] Ricardo Baeza-Yates, Di Jiang, Fabrizio Silvestri, and Beverly Harrison. Predicting the next app that you are going to use. In *Proceedings of the Eighth*

ACM International Conference on Web Search and Data Mining, pages 285–294, 2015.

[5] Saeid Balaneshinkordan and Alexander Kotov. Embedding-based query expansion for weighted sequential dependence retrieval model. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1213–1216, 2017.

[6] Paul N. Bennett, Ryen W. White, Wei Chu, Susan T. Dumais, Peter Bailey, Fedor Borisyuk, and Xiaoyuan Cui. Modeling the impact of short- and long-term behavior on search personalization. In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 185–194, 2012.

[7] Adam L. Berger and John D. Lafferty. Information retrieval as statistical translation. In *Proceedings of the 22nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 222–229, 1999.

[8] Sumit Bhatia, Debapriyo Majumdar, and Nitish Aggarwal. Proactive information retrieval: Anticipating users' information need. In *Advances in Information Retrieval - 38th European Conference on IR Research*, volume 9626, pages 874–877, 2016.

[9] Roi Blanco, Giuseppe Ottaviano, and Edgar Meij. Fast and space-efficient entity linking for queries. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*, pages 179–188, 2015.

[10] Kurt D. Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 1247–1250, 2008.

[11] Fei Cai, Shangsong Liang, and Maarten de Rijke. Time-sensitive personalized query auto-completion. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, pages 1599–1608, 2014.

[12] David Carmel, Naama Zwerdling, Ido Guy, Shila Ofek-Koifman, Nadav Har'El, Inbal Ronen, Erel Uziel, Sivan Yogev, and Sergey Chernov. Personalized social search based on the user's social network. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, pages 1227–1236, 2009.

[13] David Carmel, Ming-Wei Chang, Evgeniy Gabrilovich, Bo-June Paul Hsu, and Kuansan Wang, editors. *ERD'14, Proceedings of the First ACM International Workshop on Entity Recognition & Disambiguation*, 2014. ACM. ISBN 978-1-4503-3023-7.

[14] David Carmel, Avihai Mejer, Yuval Pinter, and Idan Szpektor. Improving term weighting for community question answering search using syntactic analysis. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, pages 351–360, 2014.

[15] Aleksandr Chuklin, Ilya Markov, and Maarten de Rijke. *Click Models for Web Search*. Synthesis Lectures on Information Concepts, Retrieval, and Services. 2015.

[16] Marco Cornolti, Paolo Ferragina, Massimiliano Ciaramita, Hinrich Schütze, and Stefan Rüd. The SMAPH system for query entity recognition and disambiguation. In *Proceedings of the First ACM International Workshop on Entity Recognition and Disambiguation*, pages 25–30, 2014.

[17] Riley Crane and Didier Sornette. Robust dynamic classes revealed by measuring the response function of a social system. *Proceedings of the National Academy of Sciences*, 105(41):15649–15653, 2008.

[18] Mariam Daoud, Lynda Tamine-Lechani, and Mohand Boughanem. Towards a graph-based user profile modeling for a session-based personalized search. *Knowl. Inf. Syst.*, 21(3):365–398, 2009.

[19] Mostafa Dehghani, Hamed Zamani, Aliaksei Severyn, Jaap Kamps, and W. Bruce Croft. Neural ranking models with weak supervision. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 65–74, 2017.

[20] Fernando Diaz, Bhaskar Mitra, and Nick Craswell. Query expansion with locally-trained word embeddings. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 367–377, 2016.

[21] Guy Divita, Tony Tse, and Laura Roth. Failure analysis of metamap transfer (mmtx). In *Proceedings of the 11th World Congress on Medical Informatics*, volume 107, pages 763–767, 2004.

[22] Anlei Dong, Yi Chang, Zhaohui Zheng, Gilad Mishne, Jing Bai, Ruiqiang Zhang, Karolina Buchner, Ciya Liao, and Fernando Diaz. Towards recency ranking in web search. In *Proceedings of the Third International Conference on Web Search and Data Mining*, pages 11–20, 2010.

[23] Carlos Escolano, Marta R. Costa-jussà, and José A. R. Fonollosa. Towards interlingua neural machine translation. *CoRR*, abs/1905.06831, 2019.

[24] David A. Ferrucci, Eric W. Brown, Jennifer Chu-Carroll, James Fan, David Gondek, Aditya Kalyanpur, Adam Lally, J. William Murdock, Eric Nyberg, John M. Prager, Nico Schlaefer, and Christopher A. Welty. Building watson: An overview of the deepqa project. *AI Magazine*, 31(3):59–79, 2010.

[25] Wei Gao, Cheng Niu, Jian-Yun Nie, Ming Zhou, Kam-Fai Wong, and Hsiao-Wuen Hon. Exploiting query logs for cross-lingual query suggestions. *ACM Trans. Inf. Syst.*, 28(2):6:1–6:33, 2010.

[26] Mihajlo Grbovic, Nemanja Djuric, Vladan Radosavljevic, and Narayan Bhamidipati. Search retargeting using directed query embeddings. In *Proceedings of the 24th International Conference on World Wide Web*, pages 37–38, 2015.

[27] Mihajlo Grbovic, Nemanja Djuric, Vladan Radosavljevic, Fabrizio Silvestri, and Narayan Bhamidipati. Context- and content-aware embeddings for query rewriting in sponsored search. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 383–392, 2015.

[28] Ido Guy. Searching by talking: Analysis of voice queries on mobile web search. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 35–44, 2016.

[29] Morgan Harvey, Fabio Crestani, and Mark James Carman. Building user profiles from topic models for personalised search. In *Proceedings of the 22nd ACM international conference on Conference on information and knowledge management*, pages 2309–2314, 2013.

[30] Homa B. Hashemi, Amir Asiaee, and Reiner Kraft. Query intern detection using convolutional neural networks. *WSDM QRUMS 2016 Workshop*, 2016.

[31] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[32] Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry P. Heck. Learning deep structured semantic models for web search using clickthrough data. In *Proceedings of the 22nd ACM International Conference on Information and Knowledge Management*, pages 2333–2338, 2013.

[33] Kevin Jarrett, Koray Kavukcuoglu, Marc'Aurelio Ranzato, and Yann LeCun. What is the best multi-stage architecture for object recognition? In *Proceedings of the 12th IEEE International Conference on Computer Vision*, pages 2146–2153, 2009.

[34] Glen Jeh and Jennifer Widom. Scaling personalized web search. In *Proceedings of the 12th International Conference on World Wide Web*, pages 271–279, 2003.

[35] Rosie Jones and Fernando Diaz. Temporal profiles of queries. *ACM Trans. Inf. Syst.*, 25(3):14, 2007.

[36] Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. A convolutional neural network for modelling sentences. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 655–665, 2014.

[37] Diane Kelly. Methods for evaluating interactive information retrieval systems with users. *Foundations and Trends in Information Retrieval*, 3(1-2):1–224, 2009.

[38] Yoon Kim. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1746–1751, 2014.

[39] Jon M. Kleinberg. Bursty and hierarchical structure in streams. *Data Min. Knowl. Discov.*, 7(4):373–397, 2003.

[40] Oleksandr Kolomiyets and Marie-Francine Moens. A survey on question answering technology from an information retrieval perspective. *Information Sciences*, 181(24):5412–5434, 2011.

[41] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Proceedings of the 26th Annual Conference on Neural Information Processing Systems*, pages 1106–1114, 2012.

[42] Anagha Kulkarni, Jaime Teevan, Krysta Marie Svore, and Susan T. Dumais. Understanding temporal query dynamics. In *Proceedings of the Forth International Conference on Web Search and Data Mining*, pages 167–176, 2011.

[43] Saar Kuzi, Anna Shtok, and Oren Kurland. Query expansion using word embeddings. In *Proceedings of the 25th ACM International Conference on Information and Knowledge Management*, pages 1929–1932, 2016.

[44] Quoc V. Le and Tomas Mikolov. Distributed representations of sentences and documents. In *Proceedings of the 31th International Conference on Machine Learning*, pages 1188–1196, 2014.

[45] Hang Li and Jun Xu. Semantic matching in search. *Foundations and Trends in Information Retrieval*, 7(5):343–469, 2014.

[46] Jiaqing Liang, Yanghua Xiao, Haixun Wang, Yi Zhang, and Wei Wang. Probase+: Inferring missing links in conceptual taxonomies. *IEEE Trans. Knowl. Data Eng.*, 29(6):1281–1295, 2017.

[47] Nut Limsopatham and Nigel Collier. Normalising medical concepts in social media texts by learning semantic representation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 1014–1023, 2016.

[48] Xiaojie Liu, Jian-Yun Nie, and Alessandro Sordoni. Constraining word embeddings by prior knowledge - application to medical information retrieval. In *Proceedings of the 12th Asia Information Retrieval Societies Conference*, volume 9994, pages 155–167, 2016.

[49] Z. Liu, Z. Niu, J.-Y. Nie, H. Wu, and H. Wang. Conversation in ir: its role and utility. In *SIGIR Workshop on Conversational Approaches to IR*, 2017.

[50] Mausam. Open information extraction systems and downstream applications. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, pages 4074–4077, 2016.

[51] Tomas Mikolov, Martin Karafiát, Lukás Burget, Jan Cernocký, and Sanjeev Khudanpur. Recurrent neural network based language model. In *Proceedings of the 11th Annual Conference of the International Speech Communication Association*, pages 1045–1048, 2010.

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

[53] Tom M. Mitchell, William W. Cohen, Estevam R. Hruschka Jr., Partha Pratim Talukdar, Justin Betteridge, Andrew Carlson, Bhavana Dalvi Mishra, Matthew Gardner, Bryan Kisiel, Jayant Krishnamurthy, Ni Lao, Kathryn Mazaitis, Thahir Mohamed, Ndapandula Nakashole, Emmanouil A. Platanios, Alan Ritter, Mehdi Samadi, Burr Settles, Richard C. Wang, Derry Wijaya, Abhinav Gupta, Xinlei Chen, Abulhair Saparov, Malcolm Greaves, and Joel Welling. Never-ending learning. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, pages 2302–2310, 2015.

[54] Bhaskar Mitra and Nick Craswell. Query auto-completion for rare prefixes. In *Proceedings of the 24th ACM International Conference on Information and Knowledge Management*, pages 1755–1758, 2015.

[55] Bhaskar Mitra, Fernando Diaz, and Nick Craswell. Learning to match using local and distributed representations of text for web search. In *Proceedings of the 26th International Conference on World Wide Web*, pages 1291–1299, 2017.

[56] Eric T. Nalisnick, Bhaskar Mitra, Nick Craswell, and Rich Caruana. Improving document ranking with dual word embeddings. In *Proceedings of the 25th International Conference on World Wide Web*, pages 83–84, 2016.

[57] Rodrigo Nogueira and Kyunghyun Cho. Task-oriented query reformulation with reinforcement learning. *CoRR*, abs/1704.04572, 2017.

[58] Hamid Palangi, Li Deng, Yelong Shen, Jianfeng Gao, Xiaodong He, Jianshu Chen, Xinying Song, and Rabab K. Ward. Deep sentence embedding using long short-term memory networks: Analysis and application to information retrieval. *IEEE/ACM Trans. Audio, Speech and Language Processing*, 24(4): 694–707, 2016.

[59] Dae Hoon Park and Rikio Chiba. A neural language model for query auto-completion. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1189–1192, 2017.

[60] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1532–1543, 2014.

[61] Francesco Piccinno and Paolo Ferragina. From tagme to WAT: a new entity annotator. In *Proceedings of the First ACM International Workshop on Entity Recognition and Disambiguation*, pages 55–62, 2014.

[62] James E. Pitkow, Hinrich Schütze, Todd A. Cass, Robert Cooley, Don Turnbull, Andy Edmonds, Eytan Adar, and Thomas M. Breuel. Personalized search. *Commun. ACM*, 45(9):50–55, 2002.

[63] Suhas Ranganath. Leveraging catalog knowledge graphs for query attribute identification in e-commerce sites. *CoRR*, abs/1807.04923, 2018.

[64] Shubhra Kanti Karmaker Santu, Liangda Li, Dae Hoon Park, Yi Chang, and ChengXiang Zhai. Modeling the influence of popular trending events on user search behavior. In *Proceedings of the 26th International Conference on World Wide Web*, pages 535–544, 2017.

[65] Ruhi Sarikaya. The technology behind personal digital assistants: An overview of the system architecture and key components. *IEEE Signal Processing Magazine*, 34(1):67–81, 2017.

[66] Denis Savenkov, Scott Weitzner, and Eugene Agichtein. Crowdsourcing for (almost) real-time question answering. In *Workshop on Human-Computer Question Answering, NAACL*, 2016.

[67] Wei Shen and Jian-Yun Nie. Is concept mapping useful for biomedical information retrieval? In *Proceedings of the 6th International Conference of the CLEF Association*, volume 9283, pages 281–286, 2015.

[68] Yelong Shen, Xiaodong He, Jianfeng Gao, Li Deng, and Grégoire Mesnil. A latent semantic model with convolutional-pooling structure for information retrieval. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, pages 101–110, 2014.

[69] Yangyang Shi, Kaisheng Yao, Le Tian, and Daxin Jiang. Deep LSTM based feature mapping for query classification. In *Proceedings of the 2016 Con-*

ference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 1501–1511, 2016.

[70] Milad Shokouhi and Kira Radinsky. Time-sensitive query auto-completion. In *The 35th International ACM SIGIR conference on research and development in Information Retrieval*, pages 601–610, 2012.

[71] Milad Shokouhi, Ryen W. White, Paul N. Bennett, and Filip Radlinski. Fighting search engine amnesia: reranking repeated results. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 273–282, 2013.

[72] Yang Song, Dengyong Zhou, and Li-wei He. Query suggestion by constructing term-transition graphs. In *Proceedings of the Fifth International Conference on Web Search and Data Mining*, pages 353–362, 2012.

[73] Alessandro Sordoni, Yoshua Bengio, and Jian-Yun Nie. Learning concept embeddings for query expansion by quantum entropy minimization. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, pages 1586–1592, 2014.

[74] Alessandro Sordoni, Yoshua Bengio, Hossein Vahabi, Christina Lioma, Jakob Grue Simonsen, and Jian-Yun Nie. A hierarchical recurrent encoder-decoder for generative context-aware query suggestion. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 553–562, 2015.

[75] Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. Yago: a core of semantic knowledge. In *Proceedings of the 16th International Conference on World Wide Web*, pages 697–706, 2007.

[76] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. In *Proceedings of the 27th Annual Conference on Neural Information Processing Systems*, pages 3104–3112, 2014.

[77] Chuanqi Tan, Furu Wei, Pengjie Ren, Weifeng Lv, and Ming Zhou. Entity linking for queries by searching wikipedia sentences. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 68–77, 2017.

[78] Robert S. Taylor. Question negotiation and information seeking in libraries. In *A. W. Elias (Ed.), (pp. 36-55) American Society for Information Science*.

[79] Jaime Teevan, Susan T. Dumais, and Eric Horvitz. Personalizing search via automated analysis of interests and activities. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 449–456, 2005.

[80] Gilad Tsur, Yuval Pinter, Idan Szpektor, and David Carmel. Identifying web queries with question intent. In *Proceedings of the 25th International Conference on World Wide Web*, pages 783–793, 2016.

[81] Ivan Vulic and Marie-Francine Moens. Monolingual and cross-lingual information retrieval models based on (bilingual) word embeddings. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 363–372, 2015.

[82] Jianqiang Wang and Douglas W. Oard. Matching meaning for cross-language information retrieval. *Information Processing and Management*, 48(4):631–653, 2012.

[83] Zhongyuan Wang, Kejun Zhao, Haixun Wang, Xiaofeng Meng, and Ji-Rong Wen. Query understanding through knowledge-based conceptualization. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence*, pages 3264–3270, 2015.

[84] Ziang Xie, Anand Avati, Naveen Arivazhagan, Dan Jurafsky, and Andrew Y. Ng. Neural language correction with character-based attention. *CoRR*, abs/1603.09727, 2016.

[85] Chenyan Xiong, Zhuyun Dai, Jamie Callan, Zhiyuan Liu, and Russell Power. End-to-end neural ad-hoc ranking with kernel pooling. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 55–64, 2017.

[86] Xing Yi and James Allan. A comparative study of utilizing topic models for information retrieval. In *Proceedings of the 31th European Conference on IR Research*, pages 29–41, 2009.

[87] Dawei Yin, Yuening Hu, Jiliang Tang, Tim Daly Jr., Mianwei Zhou, Hua Ouyang, Jianhui Chen, Changsung Kang, Hongbo Deng, Chikashi Nobata, Jean-Marc Langlois, and Yi Chang. Ranking relevance in yahoo search. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 323–332, 2016.

[88] Hamed Zamani and W. Bruce Croft. Estimating embedding vectors for queries. In *Proceedings of the 2016 ACM on International Conference on the Theory of Information Retrieval*, pages 123–132, 2016.

[89] Hamed Zamani and W. Bruce Croft. Embedding-based query language models. In *Proceedings of the 2016 ACM on International Conference on the Theory of Information Retrieval*, pages 147–156, 2016.

[90] Hamed Zamani and W. Bruce Croft. Relevance-based word embedding. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 505–514, 2017.

[91] Qing T. Zeng and Tony Tse. Exploring and developing consumer health vocabularies. *J. Am. Medical Informatics Assoc.*, 13(1):24–29, 2006.

[92] Ye Zhang and Byron C. Wallace. A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. *CoRR*, abs/1510.03820, 2015.

[93] Shi Zhao and Yan Zhang. Tailor knowledge graph for query understanding: linking intent topics by propagation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1070–1080, 2014.

[94] Guoqing Zheng and Jamie Callan. Learning to reweight terms with distributed representations. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 575–584, 2015.