Chapter 1
An Introduction to Query Understanding

Hongbo Deng and Yi Chang

Abstract This book aims to present a systematic study of practices and theories for query understanding of search engines. The studies in this book can be categorized into three major classes. One class is to figure out what the searcher wants by extracting semantic meaning from the searcher's keywords, such as query classification, query tagging, and query intent understanding. Another class is to analyze search queries and then translate them into an enhanced query that can produce better search results, such as query spelling correction, query rewriting. The third class is to assist users to refine or suggest queries so as to reduce users' search effort and satisfy their information needs, such as query auto-completion and query suggestion. This chapter discusses organization, audience, and further reading for this book.

1.1 Introduction

Query understanding is a fundamental part of search engine. It is responsible to precisely infer the intent of the query formulated by search user, to correct spelling errors in the query, to reformulate the query to capture its intent more accurately, and to guide search user in the formulation of query with precise intent. Query understanding methods generally take place before the search engine retrieves and ranks search results. If we can understand the information needs of search queries in the best way, we can better serve users. Therefore, query understanding has been recognized as the key technology for search engines.

Before we dive into the details of query understanding, let us briefly review how do search engines work. In general, search engines need to understand exactly what

H. Deng (✉)
Alibaba Group, Zhejiang, China
e-mail: hbdeng@acm.org

Y. Chang
Jilin University, Jilin, China
e-mail: yichang@jlu.edu.cn

Y. Chang, H. Deng (eds.), *Query Understanding for Search Engines*,
The Information Retrieval Series 46, https://doi.org/10.1007/978-3-030-58334-7_1

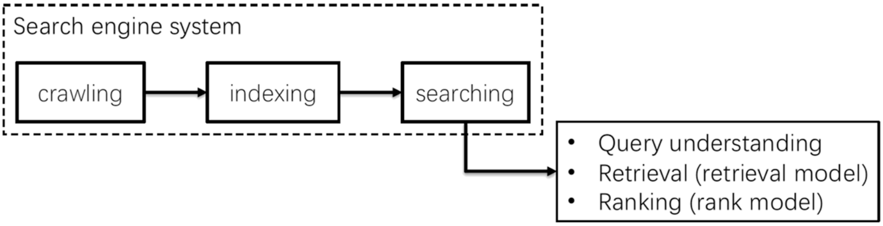

Fig. 1.1 Fundamental actions of a search engine system

kind of information is available and then present it to users logically according to their query. The way they accomplish this is through three fundamental actions: crawling, indexing, and searching, as shown in Fig. 1.1. Web search engines get their information by crawling from site to site, while some e-commerce search engines collect their information according to their own providers without crawling. The indexing process is to store and organize their collected information on their servers, as well as prepare some positive and negative signals for the following search process. Searching is a process that accepts a text query as input and returns a list of results, ranked by their relevance to the query. The search process can be further divided into three steps, which include query understanding, retrieval, and ranking.

Because query understanding is the first step in the search process, it is the core part of the process that the user interacts with search engines. The basic search interface features include query auto-completion and query refinement suggestions. More specifically, query auto-completion is becoming the primary surface for the search experience, which suggests relevant completed queries as the user types. Suppose a user wants to query "britney spears," Fig. 1.2 shows the most relevant completions for prefixes from "b" to "brit." Once a query is submitted, the primary objective is to conduct semantic analysis, so as to understand the intention behind the query, such as query classification (classifying the query according to the categories) and query tagging (extracting the entities and concepts mentioned in the query). Given the query "britney spears pretty girls" as shown in Fig. 1.2 (step 2), it can be classified to the category [music] with probability 0.5, and then "britney spears" and "pretty girls" are tagged as [singer] and [song], respectively. Before retrieving search results, another important type of query understanding is to alter a given query to alternative queries through query expansion, spelling correction, and query rewriting, which can improve relevance performance by bridging the vocabulary gap between a query and relevant documents. Much of query understanding takes place before retrieving a single result; however, some query suggestions are returned to users along with the search results, which is also called post-ranking query suggestions. It can assist users to refine queries in order to satisfy their information needs. An overview diagram of searching process is shown in Fig. 1.2.

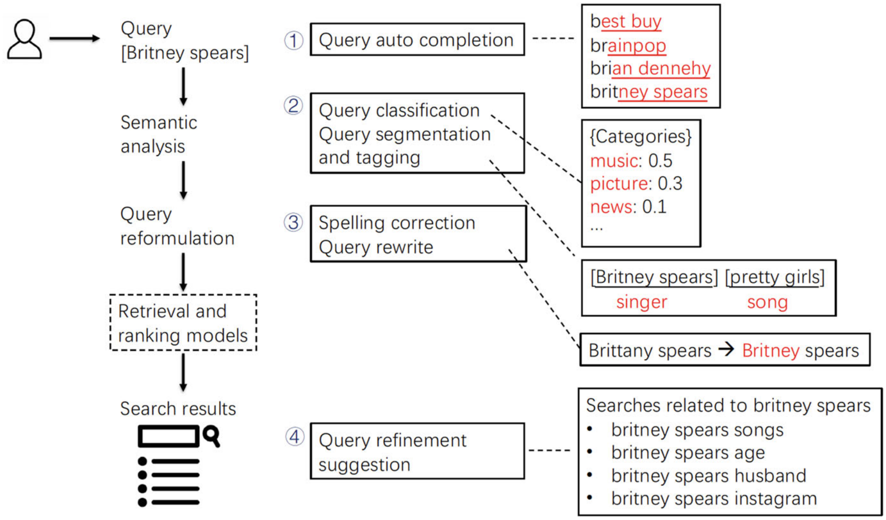

Fig. 1.2 An overview diagram of searching process

What is not covered in this book? We do not cover crawling, indexing, retrieval, and ranking problems. Some basic query processing stacks, such as stemming and lemmatization, are not covered. For more details in these areas, please refer to [12, 41].

What is covered in this book? In this book, we aim to present a systematic study of practices and theories for query understanding of search engines. This chapter will discuss how the organization of the book is related to the different areas of query understanding and will briefly discuss each of these issues in the following sections.

1.2 Query Classification

Query classification, which is to assign a search query into a given target taxonomy, has been recognized as one important technique that can bring improvements in both efficiency and effectiveness of general Web search. Various classification of taxonomies have been proposed in order to understand users' search query from different viewpoints, including intent taxonomy, topic taxonomy, performance taxonomy, and so on. Basically, these tasks of query classification include, but are not limited to, identifying the type of search goals and demanded resources required by a user, identifying the topical categories a query belongs to, determining query performance of a query for a given retrieval system, and selecting vertical services a query might be relevant to.

Query topic classification aims to determine the topical category of queries according to some predefined topic taxonomy. Typical topic taxonomies used in literature include that proposed in the KDD Cup 2005 [37] and a manually collected one from AOL [6]. Researchers may also use some specific topic taxonomy constructed in some target domains for commercial search engine. The task of KDD Cup 2005 competition was to classify 800,000 internet user search queries into 67 predefined categories. It is obvious that the KDD Cup 2005 is an important event in this area, since it provides an opportunity for researchers from different countries to develop techniques to enhance the task of query topic classification. In this competition, many methods have been proposed to tackle the main challenges of training data sparsity and feature sparsity problems. Even after the competition, there have been continuous research work to improve the performance of query topic classification. In Chap. 2, Sect. 2.3, we reviewed the representative work proposed for KDD Cup 2005 and AOL taxonomy, as well as some representative work on other topic taxonomy or specific domains.

Query performance classification aims to categorize queries according to their *difficulty*, i.e., the likely quality of results returned by the search system for a query, in the absence of relevance judgments and without user feedback. In practice, query difficulty could be further specified in two ways, namely *system query difficulty* and *collection query difficulty* [2]. System query difficulty captures the difficulty of a query for a given retrieval system run over a given collection. In other words, the query is difficult for a *particular* system. System query difficulty is typically measured by the average precision of the ranked list of documents returned by the retrieval system when run over the collection using the query in question. Collection query difficulty captures the difficulty of a query with respect to a given collection. In this way, the difficulty of a query is meant to be largely *independent* of any specific retrieval system. For more details in this direction, please refer to Chap. 2, Sect. 2.4.

In Chap. 2, we discuss several different query classification tasks, from some major interests, such as intent classification, topic classification, performance classification, to classification tasks on other "dimensions" such as geographic location and time requirement. For each classification task, there have been multiple classification taxonomies proposed in the past due to finer analysis of users' needs or specific application requests. Although different types of features have been proposed for different tasks, they are mainly from the resources such as query logs, click logs, retrieved documents, search corpus, and queries themselves. Supervised, unsupervised, and semi-supervised models have been employed in these tasks.

1.3 Query Segmentation and Tagging

Query segmentation is one of the first steps towards query understanding. Its goal is to split a query string into a few segments. The basic bag-of-words (BOW) model can be thought of as segmenting queries based on individual words. Such an approach is simple but can be less meaningful. For Chinese language, most of the individual words have little meaning by themselves and the meaning of a sentence is carried by a sequence of words. However, there are no natural boundaries such as spaces in Chinese language, and query segmentation is a necessary step for Chinese queries [44, 52] as well as for many other languages. For English language, spaces are presented inside sentences and individual words obtained in the BOW model are more meaningful compared with Chinese language. However, the BOW model can still be less effective because the meaning of a phrase can be totally different from its individual words. For example, knowing that “new york” is a city name and treating them as a whole is better than treating them as two individual words “new” and “york.” Moreover, it is also beneficial to know whether some words comprise an entity like an organization’s name, which makes it possible to enforce word proximity and ordering constraints on document matching. Therefore, it is necessary to go beyond the BOW model. A search engine that can automatically split a query into meaningful segments is highly likely to improve its overall user satisfaction.

In Chap. 3, Sect. 3.1, we formulate the problem of query segmentation as finding boundaries to segment queries into a list of semantic blocks. Various approaches have been proposed for query segmentation, which can be categorized into three different approaches, including heuristic-based approaches, supervised learning approaches, and unsupervised learning approaches. Heuristic-based approaches are based on some statistics obtained from external resources, such as pointwise mutual information (PMI) [8], connexity [46], and naive segmentation [22]. In the supervised learning setting, query segmentation is formulated as a classification problem that takes a query as input and outputs a vector with $n - 1$ binary values, where y_i means that there is a break between word x_i and x_{i+1}. Recently, a query segmentation method based on conditional random fields (CRF) is proposed by Yu et al. [58]. Supervised learning approaches rely on human annotated training data, while unsupervised learning approaches have unique advantage that no labeled data is needed. Existing approaches mainly use EM as their main algorithms. For more details about query segmentation, please refer to Chap. 3, Sect. 3.1.

The problem of query tagging is to assign labels from a set of predefined ones at word level, and it can be classified into query semantic tagging and query syntactic tagging according to different labels. One important type of semantic labels is defined along with named entity, including “Game,” “Movie,” “Book,” “Music,” etc. Given a query, the tasks of name entity recognition (NER) are to identify which words in the query represent named entities and classify them into different classes. For Web search queries, Guo et al. [21] found that only 1% of the named entity queries contain more than 1 entity and the majority of named entity queries contain exactly a single one. For e-commerce search, the semantic labels can be different properties and their values, such as brand, color, model, style, and so on. An example from [42] of query semantic tagging in the *product* domain is shown in the following where the labels are in parentheses.

> **cheap** (SortOrder) **garmin** (Brand) **steetpilot** (Model) **c340** (Model) **gps** (Type)

Semantic labels can be used to provide users with more relevant search results. For example, based on the structured information or labels generated by query tagging, many specialized search engines conduct structured matching with documents where structured information are available, such as in e-commerce search.

Another type of query tagging is related to traditional syntactic analysis, which is usually conducted over complete sentences in NLP. Its goal is to understand a sentence's grammatical constituents, POS of words, and their syntactic relations. The task of query syntactic tagging is to apply NLP techniques to search for queries. However, search queries are short and their word order is family free, which make it very challenging to directly apply syntactic parsing NLP techniques on search queries. The majority of existing approaches [5, 7, 53] transfer information from sentences in search results or snippets to search queries.

In Chap. 3, we reviewed a few representative methods for both query semantic tagging and syntactic tagging, including template-based approach, weakly supervised learning approach, transfer learning based approaches, etc.

1.4 Query Intent Understanding

Query intent itself is an ambiguous word and it is still a challenge to have a scientific definition of query intent. Intent itself means the perceived need for information that leads to a search, but how to describe or classify the need is still in an exploratory stage. Till now, different kinds of query intent understanding tasks have been explored towards discovering the implicit factors related to real user information needs according to some predefined intent taxonomy.

As a starting point, a Web search intent taxonomy with broad consensus was proposed by Broder [11], which aims to classify user goals into navigational, informational, and transactional. For instance, when a user issues the query "amazon," she could be trying to navigate the specific website http://www.amazon.com, while a user submitting "Olympic history" is most likely to be interested in finding information on that topic but not concerned about the particular website. Since the proposal of Broder's taxonomy, several other taxonomies have been proposed along the development of this area, including Rose and Levinson's taxonomy [47], Baeza-Yates's taxonomy [3], and so on.

Another well-known query intent is defined with the emergence of numerous vertical search services (e.g., job search, product search, image search, map search, news search, weather search, or academic search). Identifying the vertical intent of a given search query is becoming important in search engines to present aggregated results from multiple verticals through the standard general web search interface. This is the so-called aggregated search or universal search. For example, given the query "beijing weather," it is good to directly show the latest weather forecast information of Beijing city in the search result, while for query "tom cruise," it would be better to show the images or videos of "tom cruise" in the search result. At the same time, irrelevant vertical results within the search engine result page (SERP) may disturb users. Therefore, it is critical to have query vertical intent classifiers in a general or aggregated search engine that can predict whether a query should trigger respective vertical search services.

In Chap. 4, Sect. 4.3, we introduce the detailed query intent classification. Different classification methods have been leveraged for this task, from manual classification [30, 47] to automatical ones [3, 26, 28] such as decision tree and support vector machine. A majority of work in this area focuses on proposing effective features for query intent identification. Different kinds of features have been extracted mainly from three data resources, including search corpus, query strings, and user logs. Although the research community has consensus on the intent taxonomy, there is no standard benchmark dataset constructed for this particular task. Most researchers conducted experiments on their own labeled datasets, with query size ranging from tens to thousands.

1.5 Query Spelling Correction

Queries issued by users usually contain errors and misused words/phrases. Recent studies show that about 10 to 12% of all query terms entered into Web search engines are misspelled [16, 17]. The ability to automatically correct misspelled queries has become an indispensable component of modern search engines. Automatic spelling correction for queries helps the search engine to better understand the users' intents and can therefore improve the quality of search experience. However, query spelling is not an easy task, especially under the strict efficiency constraint. More importantly, people not only make typos on single words (insertion, deletion, and substitution), but can also easily mess up with word boundaries (concatenation and splitting). Moreover, different types of misspelling could be committed in the same query, making it even harder to correct.

Query spelling correction has long been an important research topic [29]. Traditional spellers focused on dealing with non-word errors caused by misspelling a known word as an invalid word form. Early works on query spelling correction were based on edit distance and the Trie data structure. A common strategy at that time was to utilize a trusted lexicon and certain distance measures, such as Levenshtein distance [31]. Later, noisy-channel model was introduced for spelling correction, in which the error model and n-gram language model were identified as two critical components. Brill and Moore demonstrated that a better statistical error model is crucial for improving a speller's accuracy [10]. In addition, there are many more types of spelling errors in search queries, such as misspelling, concatenation/splitting of query words, and misuse of legitimate yet inappropriate words. Research in this direction includes utilizing large web corpora and query log [1, 14, 16], training phrase-based error model from clickthrough data [54] and developing additional features [19]. More recently, [35] addressed multi-types of spelling errors using a generalized Hidden Markov Model. In Chap. 5, we will cover the detailed topics and other components for supporting a modern query spelling correction system.

1.6 Query Rewriting

It is well-known that there exists a "lexical chasm" [45] between web documents and user queries. The major reason is that web documents and user queries are created by different sets of users and they may use different vocabularies and distinct language styles. Consequently, even when the queries can perfectly match user's information needs, the search engines may be still unable to locate relevant web documents.

Query rewriting (QRW) enables the search engine to alter or expand a given query to alternative queries that can improve relevance performance by returning additional relevant results. It is a critical task in modern search engines and has attracted increasing attention in the last decade [20, 27, 45]. At the early stage, methods have been developed to find terms related to these in a given query and then substitute terms in the original queries with these related ones (or substitution-based methods) [27]. Then if we treat queries as the source language and web documents as the target language, the query rewriting problem can be naturally considered as a machine translation problem; thus, machine translation techniques have been applied for QRW (or translation-based methods) [45]. Recently, deep learning techniques have been widely applied in information retrieval [32] and natural language processing [57]. There are very recent works applying deep learning in query rewriting that achieve the state-of-the-art performance [24]. In Chap. 6, we will review the representative query rewriting methods with traditional shallow models including substitution-based methods and translation-based methods, as well as the advanced algorithms based on deep learning techniques such as word embedding, seq2seq models, learning to rewrite frameworks, and deep reinforcement learning.

1.7 Query Auto-Completion

Query auto-completion (QAC) has been widely used in all major search engines, and has become one of the most important and visible features in modern search engines. The main objective of QAC is to predict users' intended queries and assist them to formulate a query while typing. The QAC engine generally offers a list of suggested queries that start with a user's input as a prefix, and the list of suggestions is changed to match the updated input after the user types each character. The interaction with the QAC engine ends until the user clicks one of the suggestions from the list or presses return.

The most popular QAC algorithm is to suggest completions according to their past popularity. Generally, a popularity score is assigned to each query based on the frequency of the query in the query log from where the query database was built. This simple QAC algorithm is called most popular completion (MPC), which can be regarded as an approximate maximum likelihood estimator [4]. The main drawback of MPC is that it assumed user's interest is stable within the range of the collected historical query logs. However, user's interest changes from time to time and can be influenced by various types of information, including temporal information, contextual information, personal information, user's interaction in QAC, and user's interaction besides QAC. In Chap. 7, we will introduce existing metrics utilized in measuring the QAC performance and the most prominent QAC approaches in the literature, including context-sensitive QAC [4], time-sensitive QAC [49, 56], personalized QAC [48], interaction-based QAC [33, 34, 36], and so on.

1.8 Query Suggestion

Query suggestion is one of the few fundamental problems in Web search. It assists users to refine queries in order to satisfy their information needs. Most commercial search engines provide query suggestions on their search result pages to help user formulating queries. Search engine logs contain information on how users refine their queries as well as how users click on suggested queries. As a result, most query suggestion techniques leverage search logs as a useful source of information. From the perspective of modeling and organizing search logs, query suggestion techniques can be categorized into four classes: (1) query co-occurrence; (2) query-URL bipartite graph; (3) query transition graph; and (4) short-term search context methods.

In general, co-occurrence methods [18, 25, 27, 39] use co-occurrence of query pairs in sessions or tasks. This type of method is usually straight-forward to understand and compute. Query-URL bipartite graph methods [15, 43, 50] use clicked URLs of a query to find similar queries. This type of method usually conducts random walk on the click graph to propagate the similarities. Query transition graph methods [9, 51, 55] use the query refinement information in search logs to find next possible queries in the search process. This type of method usually constructs a query transition graph and performs random walk on the graph starting from testing queries. Short-term search context methods [13, 23, 25, 38, 40] use search sequence information (e.g., queries within current session) to improve the relevance of suggestions. Sequence mining approaches [13, 23, 38] are usually applied to predict next possible queries given current search sequence. In Chap. 8, we introduce the aforementioned techniques in detail and summarize other related suggestion techniques as well as future directions.

1.9 Discussion and Future Directions

The problem of query understanding has been widely studied in the Web search and data mining literature. Query understanding is not about determining the relevance of each result to the query, while it is the communication channel between the searcher and the search engine. The query understanding problem has numerous variations that allow the use of either additional domain knowledge or cross-language in order to improve the underlying results. Moreover, a wide variety of methods are available for query understanding beyond keyword query, such as natural language question understanding and dialog query conversational query understanding. With the success of deep learning in many research areas, it has started to explore deep learning based techniques to various query understanding problems, including but not limited to query classification, query tagging, query rewrite, query suggestions. In Chap. 9, we will further discuss a few other interesting cases, including personalized query understanding, temporal dynamics of queries, and semantic understanding for search queries. In many cases, these advanced techniques and algorithms may be used to significantly improve the quality of the underlying results.

References

1. Farooq Ahmad and Grzegorz Kondrak. Learning a spelling error model from search query logs. In *Proceedings of the Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 955–962, 2005.
2. Javed A. Aslam and Virgiliu Pavlu. Query hardness estimation using Jensen-Shannon divergence among multiple scoring functions. In *Proceedings of the 29th European Conference on IR Research*, volume 4425, pages 198–209, 2007.
3. Ricardo A. Baeza-Yates, Liliana Calderón-Benavides, and Cristina N. González-Caro. The intention behind web queries. In *Proceedings of the 13th International Conference on String Processing and Information Retrieval*, volume 4209, pages 98–109, 2006.
4. Ziv Bar-Yossef and Naama Kraus. Context-sensitive query auto-completion. In *Proceedings of the 20th International Conference on World Wide Web*, pages 107–116, 2011.
5. Cory Barr, Rosie Jones, and Moira Regelson. The linguistic structure of English web-search queries. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 1021–1030, 2008.

6. Steven M. Beitzel, Eric C. Jensen, Ophir Frieder, David D. Lewis, Abdur Chowdhury, and Aleksander Kolcz. Improving automatic query classification via semi-supervised learning. In *Proceedings of the 5th IEEE International Conference on Data Mining*, pages 42–49, 2005.
7. Michael Bendersky, W. Bruce Croft, and David A. Smith. Structural annotation of search queries using pseudo-relevance feedback. In *Proceedings of the 19th ACM Conference on Information and Knowledge Management*, pages 1537–1540, 2010.
8. Shane Bergsma and Qin Iris Wang. Learning noun phrase query segmentation. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 819–826, 2007.
9. Paolo Boldi, Francesco Bonchi, Carlos Castillo, Debora Donato, Aristides Gionis, and Sebastiano Vigna. The query-flow graph: model and applications. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management*, pages 609–618, 2008.
10. Eric Brill and Robert C. Moore. An improved error model for noisy channel spelling correction. In *38th Annual Meeting of the Association for Computational Linguistics*, pages 286–293, 2000.
11. Andrei Z. Broder. A taxonomy of web search. *SIGIR Forum*, 36 (2): 3–10, 2002.
12. Stefan Büttcher, Charles L. A. Clarke, and Gordon V. Cormack. *Information Retrieval - Implementing and Evaluating Search Engines*. MIT Press, 2010.
13. Huanhuan Cao, Daxin Jiang, Jian Pei, Qi He, Zhen Liao, Enhong Chen, and Hang Li. Context-aware query suggestion by mining click-through and session data. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 875–883, 2008.
14. Qing Chen, Mu Li, and Ming Zhou. Improving query spelling correction using web search results. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 181–189, 2007.
15. Nick Craswell and Martin Szummer. Random walks on the click graph. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 239–246, 2007.
16. Silviu Cucerzan and Eric Brill. Spelling correction as an iterative process that exploits the collective knowledge of web users. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 293–300, 2004.
17. Hercules Dalianis. Evaluating a spelling support in a search engine. In *Proceedings of the 6th International Conference on Applications of Natural Language to Information Systems*, volume 2553, pages 183–190, 2002.
18. Bruno M. Fonseca, Paulo Braz Golgher, Bruno Pôssas, Berthier A. Ribeiro-Neto, and Nivio Ziviani. Concept-based interactive query expansion. In *Proceedings of the 2005 ACM CIKM International Conference on Information and Knowledge Management*, pages 696–703, 2005.
19. Jianfeng Gao, Xiaolong Li, Daniel Micol, Chris Quirk, and Xu Sun. A large scale ranker-based system for search query spelling correction. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 358–366, 2010.
20. Jianfeng Gao, Shasha Xie, Xiaodong He, and Alnur Ali. Learning lexicon models from search logs for query expansion. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 666–676, 2012.
21. Jiafeng Guo, Gu Xu, Xueqi Cheng, and Hang Li. Named entity recognition in query. In *Proceedings of the 32nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 267–274, 2009.
22. Matthias Hagen, Martin Potthast, Benno Stein, and Christof Bräutigam. Query segmentation revisited. In Sadagopan Srinivasan, Krithi Ramamritham, Arun Kumar, M. P. Ravindra, Elisa Bertino, and Ravi Kumar, editors, *Proceedings of the 20th International Conference on World Wide Web*, pages 97–106, 2011.
23. Qi He, Daxin Jiang, Zhen Liao, Steven C. H. Hoi, Kuiyu Chang, Ee-Peng Lim, and Hang Li. Web query recommendation via sequential query prediction. In *Proceedings of the 25th International Conference on Data Engineering*, pages 1443–1454, 2009.

24. Yunlong He, Jiliang Tang, Hua Ouyang, Changsung Kang, Dawei Yin, and Yi Chang. Learning to rewrite queries. In *Proceedings of the 25th ACM International Conference on Information and Knowledge Management*, pages 1443–1452, 2016.
25. Chien-Kang Huang, Lee-Feng Chien, and Yen-Jen Oyang. Relevant term suggestion in interactive web search based on contextual information in query session logs. *J. Assoc. Inf. Sci. Technol.*, 54 (7): 638–649, 2003.
26. Bernard J. Jansen, Danielle L. Booth, and Amanda Spink. Determining the informational, navigational, and transactional intent of web queries. *Inf. Process. Manag.*, 44 (3): 1251–1266, 2008.
27. Rosie Jones, Benjamin Rey, Omid Madani, and Wiley Greiner. Generating query substitutions. In *Proceedings of the 15th international conference on World Wide Web*, pages 387–396, 2006.
28. In-Ho Kang and Gil-Chang Kim. Proceedings of the 26th annual international ACM SIGIR conference on research and development in information retrieval. pages 64–71, 2003.
29. Karen Kukich. Techniques for automatically correcting words in text. *ACM Computing Surveys*, 24 (4): 377–439, 1992.
30. Uichin Lee, Zhenyu Liu, and Junghoo Cho. Automatic identification of user goals in web search. In *Proceedings of the 14th international conference on World Wide Web*, pages 391–400, 2005.
31. V. I. Levenshtein. Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics Doklady.*, 10 (8): 707–710, February 1966.
32. Hang Li and Zhengdong Lu. Deep learning for information retrieval. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 1203–1206, 2016.
33. Liangda Li, Hongbo Deng, Anlei Dong, Yi Chang, Hongyuan Zha, and Ricardo Baeza-Yates. Analyzing user's sequential behavior in query auto-completion via Markov processes. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 123–132, 2015.
34. Liangda Li, Hongbo Deng, Anlei Dong, Yi Chang, Ricardo Baeza-Yates, and Hongyuan Zha. Exploring query auto-completion and click logs for contextual-aware web search and query suggestion. In *Proceedings of the 26th International Conference on World Wide Web*, pages 539–548, 2017.
35. Yanen Li, Huizhong Duan, and ChengXiang Zhai. A generalized hidden Markov model with discriminative training for query spelling correction. In *The 35th International ACM SIGIR conference on research and development in Information Retrieval*, pages 611–620, 2012.
36. Yanen Li, Anlei Dong, Hongning Wang, Hongbo Deng, Yi Chang, and ChengXiang Zhai. A two-dimensional click model for query auto-completion. In *The 37th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 455–464, 2014.
37. Ying Li, Zijian Zheng, and Honghua (Kathy) Dai. KDD CUP-2005 report: facing a great challenge. *SIGKDD Explorations*, 7 (2): 91–99, 2005.
38. Zhen Liao, Daxin Jiang, Enhong Chen, Jian Pei, Huanhuan Cao, and Hang Li. Mining concept sequences from large-scale search logs for context-aware query suggestion. *ACM Trans. Intell. Syst. Technol.*, 3 (1): 17:1–17:40, 2011.
39. Zhen Liao, Yang Song, Li-wei He, and Yalou Huang. Evaluating the effectiveness of search task trails. In *Proceedings of the 21st Conference on World Wide Web*, pages 489–498, 2012.
40. Zhen Liao, Daxin Jiang, Jian Pei, Yalou Huang, Enhong Chen, Huanhuan Cao, and Hang Li. A vlHMM approach to context-aware search. *ACM Trans. Web*, 7 (4): 22:1–22:38, 2013.
41. Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to information retrieval*. Cambridge University Press, 2008.
42. Mehdi Manshadi and Xiao Li. Semantic tagging of web search queries. In *Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 861–869, 2009.

43. Qiaozhu Mei, Dengyong Zhou, and Kenneth Ward Church. Query suggestion using hitting time. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management*, pages 469–478, 2008.
44. Fuchun Peng, Fangfang Feng, and Andrew McCallum. Chinese segmentation and new word detection using conditional random fields. In *Proceedings of the 20th International Conference on Computational Linguistics*, 2004.
45. Stefan Riezler and Yi Liu. Query rewriting using monolingual statistical machine translation. *Comput. Linguistics*, 36 (3): 569–582, 2010.
46. Knut Magne Risvik, Tomasz Mikolajewski, and Peter Boros. Query segmentation for web search. In *Proceedings of the Twelfth International World Wide Web Conference*, 2003.
47. Daniel E. Rose and Danny Levinson. Understanding user goals in web search. In *Proceedings of the 13th international conference on World Wide Web*, pages 13–19, 2004.
48. Milad Shokouhi. Learning to personalize query auto-completion. In *Proceedings of The 36th International ACM SIGIR conference on research and development in Information Retrieval*, pages 103–112, 2013.
49. Milad Shokouhi and Kira Radinsky. Time-sensitive query auto-completion. In *Proceedings of The 35th International ACM SIGIR conference on research and development in Information Retrieval*, pages 601–610, 2012.
50. Yang Song and Li-wei He. Optimal rare query suggestion with implicit user feedback. In *Proceedings of the 19th International Conference on World Wide Web*, pages 901–910, 2010.
51. 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.
52. Richard Sproat, Chilin Shih, William Gale, and Nancy Chang. A stochastic finite-state word-segmentation algorithm for Chinese. *Comput. Linguistics*, 22 (3): 377–404, 1996.
53. Xiangyan Sun, Haixun Wang, Yanghua Xiao, and Zhongyuan Wang. Syntactic parsing of web queries. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1787–1796, 2016.
54. Xu Sun, Jianfeng Gao, Daniel Micol, and Chris Quirk. Learning phrase-based spelling error models from clickthrough data. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 266–274, 2010.
55. Idan Szpektor, Aristides Gionis, and Yoelle Maarek. Improving recommendation for long-tail queries via templates. In *Proceedings of the 20th International Conference on World Wide Web*, pages 47–56, 2011.
56. Stewart Whiting and Joemon M. Jose. Recent and robust query auto-completion. In *Proceedings of the 23rd International Conference on International World Wide Web Conference*, pages 971–982, 2014.
57. Tom Young, Devamanyu Hazarika, Soujanya Poria, and Erik Cambria. Recent trends in deep learning based natural language processing [review article]. *IEEE Comput. Intell. Mag.*, 13 (3): 55–75, 2018.
58. Xiaohui Yu and Huxia Shi. Query segmentation using conditional random fields. In *Proceedings of the First International Workshop on Keyword Search on Structured Data*, pages 21–26, 2009.