Query classification using Wikipedia

Richard Khoury

Department of Software Engineering,
Lakehead University,
955 Oliver Road, Thunder Bay,
Ontario, P7B 5E1, Canada
E-mail: richard.khoury@lakehaedu.ca

Abstract: Identifying the intended topic that underlies a user's query can benefit a large range of applications, from search engines to question-answering systems. However, query classification remains a difficult challenge due to the variety of queries a user can ask, the wide range of topics users can ask about, and the limited amount of information that can be mined from the query. In this paper, we develop a new query classification system that accounts for these three challenges. Our system relies on the freely-available online encyclopaedia Wikipedia as a natural-language knowledge-based, and exploits Wikipedia's structure to infer the correct classification of any given query. We will present two variants of this query classification system in this paper, and demonstrate their reliability compared to each other and to the literature benchmarks using the query sets from the KDD CUP 2005 and TREC 2007 competitions.

Keywords: natural language processing; NLP; query classification; database systems; information retrieval systems; intelligent information systems; knowledge-based systems; web-based information systems; Wikipedia.

Reference to this paper should be made as follows: Khoury, R. (2011) 'Query classification using Wikipedia', *Int. J. Intelligent Information and Database Systems*, Vol. 5, No. 2, pp.143–163.

Biographical notes: Richard Khoury obtained his MSc and PhD in Electrical and Computer Engineering from Laval University in 2004 and from the University of Waterloo in 2007, respectively. He is currently an Assistant Professor in the Department of Software Engineering at Lakehead University. His primary area of research is natural language processing, and he has published several papers on the topic both in international journals and in peer-reviewed conference proceedings. Additional research interests include data mining, knowledge management, machine learning, computer vision, and artificial intelligence.

1 Introduction

Query classification is the task of natural language processing (NLP) whose goal is to identify the category label, in a predefined set, that best represents the domain of a question being asked. While such a categorisation task is found in several branches of NLP, the challenge of query classification is accentuated by the fact that a typical query is only a few words long and that the user's intended topic is often subjective (Li et al.,

2005). Moreover, queries can come in two different basic styles: either as complete and grammatically-correct questions of the kind a person would ask to a question-answering (QA) system or another human being, or as web-style keyword-only searches. Finally, given the growing shift in focus of the NLP community towards web queries, general classification systems now have to handle queries on any domain found on the internet, which is to say any domain at all (Jansen et al., 2000).

In this paper, we present a new query classification system which addresses all of these challenges. We decided to build our system based on the Wikipedia database, a design decision which brings a massive amount of natural language data from which to infer the user's implicit intent, as well as a set of 300,000 categories covering most domains of human knowledge at varying degrees of granularity. Such a set of labels allows our classifier to pinpoint the topic of any queries the user asks with the appropriate level of detail. One contribution in this paper is the development of a mathematical framework for that system, which can make accurate classifications based on the very large but simple natural language corpus we extracted from Wikipedia. In addition, a second interesting contribution is the study of two variants of our query classifier, which differ only on which information is retained from the Wikipedia articles. The conclusions we draw from that study can help guide the design and development of other Wikipedia-based and encyclopaedia-based NLP systems.

The rest of the paper is organised as follows. Section 2 presents overviews of the literature both in the field of query classification and on the more general topic of NLP research using Wikipedia. We present in detail both variants of our classification system in Section 3, then we move on in Section 4 to describe two sets of experiments performed on this system and to analyse their results. Finally, we give some concluding remarks in Section 5.

2 Related work

2.1 Query classification

Many NLP applications can benefit from gaining some domain information on the text being processed. This information can be useful in several tasks, such as word disambiguation, keyword and keyphrase identification, and categorisation. Query classification is the task of NLP that focuses on inferring the domain information surrounding user-written queries, and on assigning each query to the category label that best represents its domain in a predefined set of labels. We should distinguish this task from that of question type classification, which consists in inferring the type of information asked about in the query (a person, a place, a date, a definition, and so on), and which is outside the scope of this research. Recently, web queries have received particular attention, both for their abundance [users make around 10 billion queries per month (Hu et al., 2009)] and for their market value. However, the scarcity of information in a web query presents a sizeable challenge. While traditional NLP systems can use information ranging from multi-word windows to entire documents, the typical query is very short. Some research has shown that 62% of queries feature two terms or less (Jansen et al., 2000), while 79% of queries used in the ACM KDD CUP 2005 competition featured four terms or less (Shen et al., 2005). Moreover, other sources of information often used in NLP are unreliable when it comes to queries. For example,

search history is useless because about 66% of users only submit one query per search session, and relevance feedback on the results is given half as often by users in internet search as it is in other information retrieval systems (Jansen et al., 2000). Despite these difficulties, query classification remains an active field of research. Indeed, an accurate query classification system has vast potential applications ranging from superior web search tools (Hu et al., 2009) to improved QA tools (Fu et al., 2009).

Given the sparsity of information that can be gathered directly from a query, many researchers design query classification systems that rely on an outside knowledge source. One popular source, when available, is a domain ontology. For example, Fu et al. (2009) use a music knowledge ontology of 54 classes in their research. Ontology keywords are identified in the query, and the classification is done by computing the Bayesian probability of each ontology class given the keywords found. However, such a method can only be considered when an appropriate domain ontology already exists, and that is not always the case. In their absence, researchers have had to create their own knowledge bases from various sources. For instance, Beitzel et al. (2005) use a database of 20,000 manually-classified web queries as a knowledge base. To classify a query, they begin by parsing it to extract syntactic relationships and match these relationships to those in the database, and then compute the selectional preference of the parsed query to find the best category to classify it in. In an alternative approach, the query classification system proposed by Jingbo and Na (2008) relies on a domain-specific knowledge base automatically constructed from a corpus of web pages and some domain-specific seed keywords. First, the web pages are classified in the appropriate domain given the occurrence of seed keywords. Then the query's word vector is compared to the vectors of the websites using a cosine measure, and the query is classified in the same domain as the website to which it was most similar. A comparable approach is adopted by Shen et al. (2005), a main difference being that they construct their corpus from web pages already categorised in the Google directory, and thus, eliminate the need for the keyword-based web page classification step present in Jingbo and Na (2008). One last interesting sample of this type of project is that of Hu et al. (2009), who use the Wikipedia category and article graphs as a knowledge base. More specifically, given some seed concepts they want their query classification system to recognise; they target the relevant articles and categories and construct a graph of Wikipedia domains by following the links in these articles using a Markov random walk algorithm. Each step from one concept to the next on the graph is assigned a transition probability, and these probabilities are then used to compute the likelihood of each domain. Once the knowledge base has been built in this way, a new user query can be classified simply by using its keywords to retrieve a list of relevant Wikipedia domains, and sorting them by likelihood. Unfortunately, their system remained small-scale and limited to only three basic domains, namely 'travel', 'personal name' and 'job'.

These projects illustrate how researchers counter the limited amount of information in the query itself with various knowledge bases, from smaller domain-specific ontologies to larger general website directories. To make the classification systems more reliable and give them a wider coverage of domains and a more detailed division inside each domain, larger and more complete knowledge bases are necessary. By these standards, Wikipedia stands out as an excellent knowledge base thanks to its width and depth of domain coverage (Khoury, 2009). For that reason, it has already been used in a number of NLP projects, including document classification (Schönhofen, 2006) and question

answering (Ahn et al., 2005). But for query classification, it seems that Hu et al. (2009) claim to have built the first system based on Wikipedia's database.

2.2 Wikipedia in NLP

Since its creation in 2001, Wikipedia[1], 'the free encyclopaedia anyone can edit', has grown in popularity to become one of the most visited and cited websites on the internet. Indeed, the Alexa[2] traffic ratings places Wikipedia in the top-ten most visited websites on the internet, and its articles commonly appear as the top results of popular PageRank-based search engines. Whereas, only a few years ago citing Wikipedia as a source of information would have seemed strange and eccentric, today it is commonplace to find it listed as background information for news stories and, unfortunately, undergraduate student papers (Viégas et al., 2007). Moreover, a number of NLP researchers have independently started using Wikipedia as a knowledge source for applications in a large range of NLP classification projects.

We have already presented how Hu et al. (2009) exploited Wikipedia to build a query classification system. An interesting related challenge is that of document classification, or of assigning documents to a predefined set of category labels. Schönhofen (2006) developed a system to accomplish this using Wikipedia's categories as labels. Schönhofen's document classification system begins by building a list of titles that are 'supported' by the document, in the sense that no more than one title word is missing from the document's text. The titles are matched to their corresponding articles, and each article contains a list of categories. Each category is then weighted according to how many articles point to it, how many titles point to these articles, the number of words in these titles, and the tf.idf value of these words, and the top results are returned. Another NLP task closely related to query classification is that of automated QA. Ahn et al. (2005) designed a system to gather the information needed to answer a user's query from Wikipedia. Their system begins by searching Wikipedia for an article related to the question. It then scans the article for named entities and assigns more importance to those occurring early in the article or in sentences that are similar to the question, and returns the most important one as the answer. Another popular NLP classification task for which Wikipedia is becoming increasingly popular is that of named entity disambiguation. Mihalcea (2007) proposed a simple system to accomplish this goal. It begins by extracting wikilinks from articles. A typical wikilink can look like '[[bar (law)|bar]]', where the right-hand side is a potentially-ambiguous word ('bar') and the left-hand side is the unambiguous entity intended ('bar (law)', the article for the legal meaning of the word). Mihalcea gathered wikilinks for 49 such ambiguous nouns, and created a set of unambiguous keywords to use to distinguish between the different entities using the left-hand part of the wikilink and the WordNet synset corresponding to the entity. A more sophisticated named entity disambiguation system was proposed by Cucerzan (2007). His system automatically built a list of pairings of ambiguous words and entities by using both sides of wikilinks as well as page titles. It then gathered the context of each entity from its matching article. This context is composed of the category list at the end of the article, the words in parenthesis in the article title, and wikilink words that recurred frequently in the article text. An ambiguous word encountered in a text can then be classified to its correct entity by comparing the text to the entity's context.

The final NLP system we would like to consider here is not directly related to our work, but is interesting for the simplicity and effectiveness of its solution. It deals with the challenge of computing the similarity between synonymous words. This is a necessary task when dealing with even simple text understanding problems, such as recognising that a sentence about 'buying a car' and one about 'acquiring an automobile' actually refer to the same topic. Towards that end, Wee and Hassan (2008) experimented with a method of computing the similarity of a new word w_1 to a known word w_2, based on the ratio of the number of Wikipedia articles in which both words are encountered to the total number of articles in which word w_2 appears. Their experimental results show that their metric yields a 9% improvement in accuracy over the next-best performing algorithm in the literature. They credit this gain in part to the fact that, thanks to Wikipedia's sheer size, their algorithm could compute the similarity between many more pairs of words than other algorithms that were based on more limited semantic resources. In making that last statement, Wee and Hassan explicitly stated the most often-cited reason in the literature to use Wikipedia as a knowledge base, namely its sheer size and width and depth of coverage (Ahn et al., 2005; Schönhofen, 2006; Wee and Hassan, 2008; Mihalcea, 2007). Many NLP applications require access to a large knowledge base, and while a great number of encyclopaedias, both general and domain-specific, are available to researchers today to help them in these projects, Wikipedia is orders of magnitude larger than all but a few extremely specialised resources (Khoury, 2009), (Voss, 2005). Moreover, Wikipedia is growing exponentially in all aspects, including the number of articles, the size of the articles, and the number of active editors (Voss, 2005). Another, often understated advantage of using Wikipedia rather than another encyclopaedia is that the source code of Wikipedia articles makes heavy use of Wiki mark-up tags, which makes them a lot easier than free text for an automated system to handle (Schönhofen, 2006). To be sure, there are also downsides to using Wikipedia, such as the presence of inaccurate information and vandalism in the articles; but this does not significantly affect NLP systems such as ours. A more serious issue is that the coverage of Wikipedia's articles and categories is heavily biased towards popular topics and recent events, unlike more balanced traditional encyclopaedias (Khoury, 2009). However, we believe that the sheer extent of Wikipedia's coverage makes up for this imbalance.

3 Methodology

The novel query classifier we propose in this paper is based on a simple database of natural language information collected from an online encyclopaedia; in this research, Wikipedia specifically. To set ideas, we will begin in Section 3.1 by describing the processing steps needed to extract the needed information from Wikipedia and structure it properly. While the resulting corpus is fairly simple, we believe this simplicity is an advantage for the overall system, as it makes it possible to create the entire database in a matter of hours instead of being a large and ongoing research project (Auer et al., 2007) (Völkel et al., 2006), and leaves open the possibility of future expansions. After discussing the corpus, we then move on in Section 3.2 to describe our classification algorithm, which is the main focus of this paper.

3.1 Corpus preparation

The Wikipedia corpus can be downloaded freely from the Wikimedia Foundation. A snapshot of Wikipedia is made available periodically; this research was done using the version from March 2008. In its original form, the corpus is a single XML file 18 GB in size, containing the articles written in the Wiki mark-up language as well as additional information available on and about the encyclopaedia. Before we can work efficiently with the corpus, we have to make it undergo a set of preparation steps. These steps are presented here, in conceptual order rather than algorithmic order for ease of comprehension.

To set ideas, we will begin by defining a *title* in Wikipedia as the unique name of a page. For our purposes, we can define three types of pages: articles, redirects, and disambiguations, and each page has a unique article title, redirect title, and disambiguation title, respectively. An *article* is an encyclopaedic entry on a topic, and the article title is its subject matter's most common name (per Wikipedia policy). For example, the article about the USA appears under the title 'United States'. Other common names for a subject, including acronyms and typos, are *redirect titles*, and the corresponding *redirect* page contains a single line usually linking to the article title. For example, the page 'USA' is a redirect to 'United States'. On the other hand, when an ambiguous name can refer to several different subjects, it becomes a *disambiguation* page whose sole purpose is to list all titles using that name. For example, 'America' is a *disambiguation title* whose corresponding page lists, among others, the title of 'the Americas', the 'United States of America', the ship 'USS America', and the actress 'America Ferrera'.

The first of our preparation steps consists simply in filtering out all the data in the Wikipedia snapshot except for titles and their pages. Because the corpus uses clear and unambiguous XML tags, this step can be easily accomplished in a single sequential scan.

The next step is title processing. We begin by performing stopword removal and stemming on the titles, and we delete empty titles that were originally composed only of stopwords. Then we insure that each title points only to articles. For redirect titles, we follow the redirect links from page to page until we reach an article, and we delete the intermediary steps. For disambiguation titles, we apply the same process to each title listed on the disambiguation page. After the completion of this title processing step, we have a many-to-many relationship between titles and articles. Indeed, titles that only differed in their stopwords or capitalisation will collapse into a single title pointing to several articles, and disambiguation titles will point to multiple articles instead of a single page listing multiple titles. On the other hand, redirect titles for a given subject will now all point to that article instead of individual redirect pages.

The third preparation step is article processing. This step begins by detecting and filtering out articles that are not encyclopaedic topics. These are mainly articles necessary for Wikipedia's management and functioning, such as help pages, user pages, and talk pages. However, we also filter out some actual articles, such as the 'list of' pages and the 7,000 UN locode pages. When an article is filtered out, the link from its title is also deleted, and if the title is orphaned it is filtered out as well. Next, we separate the list of categories present at the end of the article from the rest of the article text. As we mentioned in the introduction, we consider in this paper two variants of our query classification system; the difference between them is introduced at this point. Indeed, the two variants follow from two definitions we experimented with for what constitutes the

article text. In the first variant, we kept the entire content of the article, while in the second we restricted it only to the text inside the wikilinks' double-square-bracket mark-up code. The potential impact of these two definitions will become clear in the next section. For now, both variants of the article text undergo standard NLP treatment: stopword removal, stemming, and deleting the Wiki mark-up tags. The categories are stripped only of their Wiki mark-up tags and are passed to the next processing step.

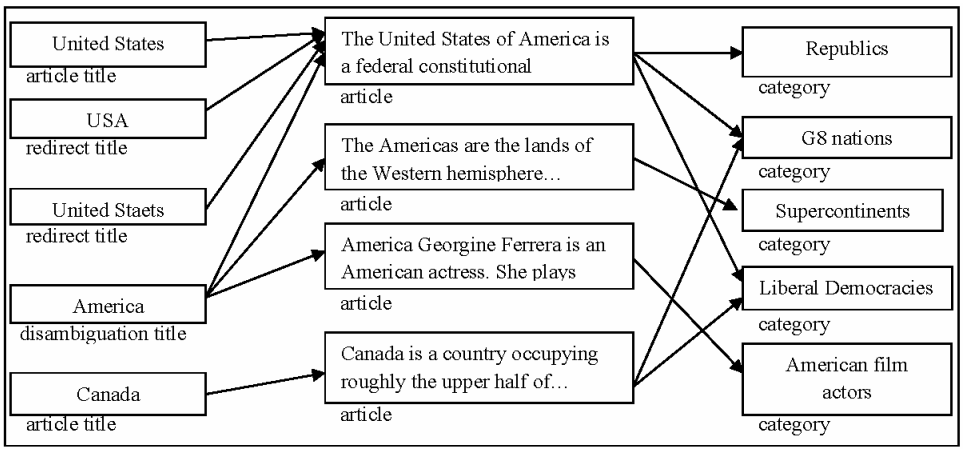
The fourth and final processing step is category processing. As mentioned above, this step receives as input the list of categories of each article. Since the ultimate goal of our system will be to classify user queries in these categories, we can filter out categories that are semantically meaningless for that purpose. These include categories meant for Wikipedia administration, such as the 'protected articles' and the 'articles lacking references' categories, as well as overly general categories such as 'living people', 'people from' individual cities, and general events by years such as '1920s births' or '1873 establishments'. Finally, 'stub' categories are merged with their real categories; e.g., the 'literature stub' category is merged with 'literature'. Once the set of categories is filtered, we define each *category vocabulary* set as the words of the article titles of all articles pointing to it, and excluding the redirect titles and disambiguation titles of these same articles. This definition is reasonable given our previous observation, that an article title is its subject matter's most common name, and therefore the most significant name from a human point of view. The category's vocabulary is therefore the set of most significant words associated with topics in that category.

At the end of this processing, the resulting Wikipedia corpus is composed of approximately 4 million titles, 2 million articles and 300,000 categories. The variant of the system that kept the entire content of articles catalogues over 3 million words, while the one restricted to words inside wikilinks only features 1.5 million words. There is a many-to-many relationship between titles, articles and categories. As we mentioned, each title can point to several different articles, and each article can be pointed to by several different titles. Likewise, each article can point to several different categories, and each category can be pointed to by several different articles. An illustrative example of this structure is given in Figure 1. Note that, for clarity, this example does not include the text stemming and stopword removal steps.

To be sure, ours is not the only project that involves extracting and stitching together information taken from Wikipedia. In fact, doing so can be the focus of entire research projects, such as DBpedia (Auer et al., 2007) or Semantic MediaWiki (Völkel et al., 2006). Our system is not incompatible with such projects; quite the opposite in fact. Such systems extract and identify a larger range of information than is found in our current corpus, and as we will indicate in our concluding remarks in Section 5 the addition of more information into our system and the study of its impact is one of the ongoing focuses of our research. In this initial stage of the project, however, we opted for the simpler corpus extraction and infrastructure we described in this section, for several reasons. First of all, this shifts the challenge of the work to developing a good set of classification equations which can work with sparse semantic information, which we describe in the next section, rather than the challenge of identifying and tagging as much semantic information as possible from the initial data. Such equations will then be easy to expand to take into account new sources of semantic information. And second, the challenge of tagging semantic information from Wikipedia is an ongoing one, and is far from being solved. Each different system takes its own unique approach, and a single

accepted standard is slow in emerging – e.g., the creators of Semantic MediaWiki had 'strong reasons to believe' that their system would be adopted into the English version of Wikipedia 'by the end of 2006' (Völkel et al., 2006), a prediction that has not been realised four years later. In this context, we find it preferable to base our system on a simple encyclopaedic corpus that could later be substituted for a richer resource, rather than tie it right away to a specific semantically-tagged version of Wikipedia. On the same topic, we can note that, while our system represents titles, articles and category vocabularies as simple bags-of-words, other semantically-richer representations are not incompatible with our work. For example, a lot of promising work has been done on the topic of explicit semantic analysis (ESA) using Wikipedia (Gabrilovich and Markovitch, 2007). This representation makes it possible to recognise the semantic relationship between words, such as that between 'equipment' and 'tool'. Enriching our text representation in this way could be beneficial, to improve the handling of queries by understanding 'what the user meant' rather than strictly what they wrote. As we explained above, in this initial stage of the project we opted to leave our corpus deliberately simple, but the addition of an ESA component could be interesting to study in follow-up research.

Figure 1 Example of the structure of the Wikipedia corpus after preparation



It is important to note that, while we specifically used Wikipedia in this study, the corpus preparation method described above could be generalised for other online encyclopaedia. Our only requirement is that the encyclopaedia structure must follow the title-article-category structure, which is not a restrictive constraint, and it is possible to restructure it to fit our needs. Likewise, the classification algorithm, we present next only requires that the corpus be structured as in Figure 1. In short, our method could be generalised to work with any online encyclopaedia.

3.2 Query classification

The aim of our query classification algorithm is to assign any user's query to the category that best represents its topic in the set extracted from Wikipedia in Section 3.1. As we explained in Section 2.2, the wide range and depth of coverage of Wikipedia's category

graph should allow our system to recognise queries on practically any subject. Our classification algorithm is designed to exploit the structure of our prepared corpus, illustrated in Figure 1, in a step-by-step manner, going from the query words to titles, articles, and finally categories. A brief overview of the algorithm's steps is presented in Table 1, and the rest of this section presents them in detail. It is worth noting that a development set of a dozen queries was used to run tests while building the system. The small number of queries made the set more manageable and allowed us to keep a close eye on the impact of each change we implemented. The results obtained with this development set are referenced in this section.

Table 1 Summary of the steps of our classification algorithm

1	Query to words: Remove stopwords and perform stemming, remove words that are not part of the corpus. Weight the words of the query based on their significance in the corpus.
2	Words to titles: Select all titles that feature at least one query word. Weight titles based on their words.
3	Titles to articles: Expand the list of articles pointed to by each title, and keep only title-article pairs featuring all or most query words.
4	Titles to articles: Weight articles based on their titles.
5	Articles to categories: Expand the list of categories pointed to by each article. Weight each category based on its articles.
6	Categories to results: Normalise the category weights and return the top-ranked category.

In the first step of our algorithm, we begin by submitting the user's query to stopword removal and stemming, as we did for the Wikipedia corpus, and we then filter out words in the query that are not part of the corpus. Given that our prepared corpus is based on Wikipedia and thus, includes scientific and technical terms, proper names, abbreviations, camel case, and common typos, the number of words lost to this filtering is minimal and the words filtered out are mostly gibberish. Next, we assign a weight R_w to each word w in the query using a modified tf.idf formula:

$$R_w = \frac{1}{3}\left[\ln\left(\frac{N_t}{W_t}\right) + \ln\left(\frac{N_a}{W_a}\right) + \ln\left(\frac{N_c}{W_c}\right)\right] \tag{1}$$

In equation (1), N_t, N_a, and N_c are respectively the number of titles, articles and categories in our corpus, while W_t, W_a, and W_c are respectively the number of titles, articles and category vocabularies featuring word w. The weight computed is thus a measure of the significance of word w in Wikipedia: a word that occurs seldom will have a high value, while a word that occurs frequently will have a lower value. An in-depth study of the values of R_w is included in our discussion, in Section 4.3.

The next step is to gather the set of titles that feature at least one of the query words. We compute the weight R_t of each of these titles as a sum of the weights of the query words it features, using equation (2). Given that R_w is a measure of the significance of the query words in Wikipedia, then R_t is a measure of the significance in Wikipedia of the title as a bag of words, given the complete query.

$$R_t = \sum_w \frac{R_w \times f(w,t)}{L_Q} \tag{2}$$

In equation (2), L_Q is the length in number of words of the user's query, and $f(w, t)$ is a binary function defined as:

$$f(w,t) = \begin{cases} 1 & \text{if word } w \text{ occurs in title } t \\ 0 & \text{if word } w \text{ does not occur in title } t \end{cases} \quad (3)$$

Another way of understanding equation (2) is to see it as the sum of R_w of all query words appearing in a title, multiplied by the ratio $f(w, t) / L_Q$ which is constant for a given title. This ratio makes the title weight function of the proportion of query words that appear in the title. A title that features all query words will have the maximum ratio of 1, while a title missing words will be multiplied by a lower ratio. Multiplying this ratio by the sum of R_w insures that, given two titles featuring the same proportion of query words, the title with the more significant words will have the higher weight. In fact, given a title featuring a higher proportion of less significant query words and one with a lower proportion of more significant query words, the latter one can potentially get the higher R_t score. During the development of the system, we also considered variants of equation (2) that included a function of the proportion of words in the title that are query words. However, we found that this addition invariably biased the system in favour of short titles that feature only query words, against longer titles that featured as many or more query words as well as other words. For example, consider our development query 'which emperor was defeated at Waterloo' and two titles featuring only one query word each, 'Waterloo' and 'Battle of Waterloo'. Equation (2) ranks both these titles equally, while adding a consideration of the proportion of title words in the query gives the title 'Waterloo' (which is composed of 100% query words) twice the score of 'Battle of Waterloo' (which has 50% query words and 50% non-query words, after stopword removal). We also considered taking into account the length in words of the title in equation (2). However, Wikipedia titles are often artificially lengthened by adding descriptive words in parenthesis to differentiate similarly-named topics: e.g., compare the titles 'The cure', 'The cure (1915 film)', 'The cure (1995 film)', 'The cure (album)', 'The cure (Fringe episode)' and 'The cure (X-Men episode)'. This makes the title word count an inaccurate measure.

In the third step, we use the first connections in our prepared Wikipedia corpus going from the titles to the articles. We follow these links from each of the titles we selected to every article it points to and generate an exhaustive list of title-article pairs. We then filter these pairs using the criterion that the relevant pairs should feature most or all of the query words. Mathematically, we impose that the title-article pair should feature all query words if the query is four words long or less, and all-but-the-least-significant-word if the query has five words or more, where the least significant word is simply the one with the lowest R_w weight. The reason we have more relaxed criteria for longer queries is that, as explained in Section 2.1, queries are typically only two to four words long. When we encounter a longer query, we assume that it contains extraneous information, and that this information is the least significant keyword in the query, which we therefore exclude from our filtering criteria. Moreover, regardless of the length of the query, if all title-article pairs are removed by this filtering, the criteria are relaxed by iteratively removing the next least significant word until some title-article pairs are kept. In theory, this means that the system could iteratively reduce the query until it keeps title-article pairs that feature only the single most significant query word. But in practice, the system

rarely had to execute this iterative relaxation step, and never had to eliminate more than a single word.

We can see at this point the impact of our two definitions of what constitutes an article text, which we presented in the previous section. On the one hand, keeping the entire article's content means that the text can be thousands of words long, which will make the filtering more inclusive but more noisy. On the other hand, keeping only the wikilinks limits the number of words available for this filtering, and causes a much higher rejection rate. However, the words inside the wikilinks are the most important and significant words in the article, by Wikipedia editorial convention, and therefore, the few title-article pairs kept by this more selective filtering should be the most relevant pairs. Thus, both alternatives are justifiable in theory. Experimentally determining which of these two variants is preferable is one of the objectives of Section 4.

The filtering done in this third step is very important. Given the massive size of the Wikipedia corpus we used, any query is guaranteed to retrieve a proportion of irrelevant title-article pairs. Moreover, an irrelevant title with a high R_t weight can lead to a high value for its matching article, as we will see in equation (4), and several of them can then combine to give a high value to a popular but irrelevant category in equation (5). Filtering by forcing all query words to appear in the title or the article allows our system to detect and discard most of these irrelevant results. For example, our development query 'which emperor was defeated at Waterloo' was classified in 'Chinese Emperors' without filtering, while after filtering this category was not part of the results at all.

As we mentioned, when no title-article pairs are retained at this step, we iteratively remove the word with the lowest R_w score and try again. This situation happens for about 20% of queries, in cases where users phrase their queries using some adjectives whereas the Wikipedia article uses some synonyms. One such query is 'what is the primary symptom of a cataract', which could not be classified because no title-article pair features the words 'primary', 'symptom' and 'cataract' together. By targeting the query word with the lowest R_w, given our equation (1), we are eliminating a common adjective rather than an important and discriminating keyword. In our example, that lowest R_w word is 'primary' with a weight of 5.9, compared to 'symptom' and 'cataract' with 8.6 and 9.6 respectively. That is indeed a word that can be safely ignored without changing the meaning of the query, and by doing so the query is classified in the category 'ophthalmology'.

The fourth step assigns a weight R_a to each article, which represents the article's significance given the titles pointing to it. This weight is simply the maximum weight from all the titles that point to it, as shown in equation (4). We also experimented with summing the weights instead of taking the maximum, but we found this to be an unreliable measure because the number of titles pointing to an article is not a metric of the article's importance or relevance, but simply an artefact of Wikipedia's structure. As we pointed out earlier, titles can be anything, including abbreviations, acronyms, and typos. Clearly, an article's importance is not a function of how many typos people commonly make when writing it! In that respect, our observation echoes a similar conclusion reached by Schönhofen (2006). In more practical terms, when we studied the impact of summing the R_t values to compute R_a, we would often encounter situations where a given title pointed to multiple articles. However, some of these articles only had a few titles pointing to them, or even only that one, and thus, had a low value of R_a, while others had multiple titles pointing to them and their R_a value summed up to considerably

larger totals. This skewing of the results in favour of articles with multiple different but equivalent names or spellings is what we are avoiding by taking the maximum instead of the summation in equation (4).

$$R_a = \max_t R_t \tag{4}$$

Another metric that we considered for equation (4) is a function of the size of the article, either in number of words or in number of categories it points to. However, this size is not a measure of the importance or significance of the article, but simply of how much time volunteer editors spent working on it (Khoury, 2009). Moreover, the size of an article in words can range from several thousands for well-developed articles to less than a dozen for newly-created 'stub' articles, so using it introduces an unpredictable variation of several orders of magnitude in the results. For the same reason, including a measure of the occurrence frequency of query words in an article is unreliable. For example, a tf.idf formula would estimate that a given keyword is ten times more important when it occurs once in a ten-word-long stub compared to when it occurs 50 times in a 5,000-word-long article.

The fifth step is to compute the weight R_c of each category pointed to by the articles, to determine the significance of a category given the set and value of significant articles selected by the query. This category weight is defined as the sum of the weight of each article pointing to it as in equation (5). The reason we use a sum rather than a maximum as we did in equation (4) is that the number of different articles is meaningful. Indeed, as we explained in Section 3.1, each article represents a distinct topic. While the number of titles pointing to an article in equation (4) simply represents the number of different ways to name that topic, the number of articles pointing to a category represents how many different topics significantly related to the query belong to that category subject.

$$R_c = \sum_a R_a \tag{5}$$

We should be careful not to take our discussion on the relationship between articles and categories too far. While the number of relevant articles selected by our algorithm that are featured in a given category is a good measure of the relevance of the category, the total number of articles in the category is not. A category can contain a small number of articles for several very different reasons, including being a highly specialised category, an extremely general category (thus, most articles belong to more specific descendent categories), a newly-created category, a category subdivided in several fine-grained subcategories, or simply a category for an obscure topic. In our study of our development queries, we find that the correct categories our queries need to be classified into can range from three to 161 articles in size (in addition to an outlier category that counted 12,651 articles), while high-ranking incorrect categories that must be avoided range from six to 248 articles in size. This large range and overlap in values illustrates how un-discriminating the category size can be. For the same reason, a measure of the size of the category vocabulary should be avoided.

The final step of our algorithm is simply one of data presentation. Once all the categories selected by the algorithm have been assigned their weight R_c, they are normalised by dividing by the maximum weight value encountered, to generate a significance score between 0 and 1. The category or categories with a score of 1 are

returned as the classification result. In addition, a complete list of categories in decreasing order can be generated, to give more complete search results if needed.

4 Experimental results

In order to thoroughly test our query classification system, we ran two independent sets of tests. These tests are meant to reflect the two main types of questions that can be asked by users; namely grammatically-complete and correct English questions and keyword-based web queries. We used standard and publicly-available question corpora for each of these tests. For the keyword-based test, we used the KDD CUP 2005 corpus of web queries, and for the test on grammatically-complete questions, we used the set of questions from the TREC 2007 QA track.

4.1 KDD CUP 2005

KDD CUP is the ACM annual data mining and knowledge discovery competition, which each year focuses on a different challenge in this large and ever-evolving field. In 2005, the topic of the competition was 'internet user search query categorisation', and a complete technical report on the competition and its outcome was prepared by Li et al. (2005). Participants in the competition had to classify 800,000 real web search queries into a maximum of five categories taken from a set of 67 categories designed by the competition organisers to broadly cover most topics found on the internet. By the competition deadline, 32 teams had submitted 37 solutions. To evaluate these solutions, the organisers picked a subset of 800 non-junk English queries and had them classified manually by three human labellers. They then ranked the 37 solutions based on overall precision and overall F1 value, as computed by equations (6) to (10). The competition's Performance Award was given to the system with the top overall F1 value, and the Precision Award was given to the system with the top overall precision value within the top ten systems evaluated on overall F1 value. Note that participants had the option to enter their system for precision ranking but not F1 ranking or vice-versa rather than both precision and F1 ranking, and several participants chose to use that option. Consequently, the top ten systems on F1 value ranked for precision are not the same as the top ten systems ranked for F1 value.

$$\text{Precision} = \frac{\Sigma_i \text{ Number of queries correctly labelled as } c_i}{\Sigma_i \text{ Number of queries labelled as } c_i} \tag{6}$$

$$\text{Recall} = \frac{\Sigma_i \text{ Number of queries correctly labelled as } c_i}{\Sigma_i \text{ Number of queries belonging to } c_i} \tag{7}$$

$$\text{F1} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{8}$$

$$\text{Overall Precision} = \frac{1}{3}\sum_{j=1}^{3} \text{Precision against labeller } j \tag{9}$$

$$\text{Overall F1} = \frac{1}{3}\sum_{j=1}^{3} \text{F1 against labeller } j \tag{10}$$

In order for our system to compare to the KDD CUP competition results, we need to use the same set of category labels and to implement the constraint of having a maximum of five categories per query. Our system uses a considerably larger and more detailed set of 300,000 categories, and the set of KDD CUP test queries were classified into approximately 3,500 of these categories. However, the KDD CUP category set is designed to hierarchically classify any query into one of seven broad super-categories, then into one of the sub-categories of that super-category, and each super-category includes a catch-all 'other' sub-category. It is therefore possible to map each of our categories to at least one KDD CUP category, and to a maximum of three categories to allow for Wikipedia categories that cover more than one KDD CUP category. We created this mapping manually for each of the 3,500 categories used by our system to classify the 800 KDD CUP test queries and then automatically limited the category set of each query to the five most frequently-occurring mapped categories. The mapping was done on the list of categories alone without any reference to the query that was classified in it, to avoid any bias. This was a straightforward, blind mapping that could be done by a simple look-up function, but in the absence of such a function we opted to do it manually. With the mapping done, we computed the overall precision and F1 of both variants of our classification system following the KDD CUP guidelines. Our results are presented in Table 2 along with the KDD CUP mean and median, the best system on precision, the best system on F1, and the worst system overall for comparison. Note that the precision rankings greater than rank ten in that table are extrapolated from the data in Li et al. (2005); the competition only ranked systems on precision up to rank ten. As we can see from these results, both variants of our system perform quite well. Our system shows an average 12.6% improvement in precision over the competition mean, and an average 5.2% improvement in F1 over the competition median. In the rankings, we made the top ten for F1 value, which means our system performed better than three-quarters of those submitted to the competition, and we got the second and third place for precision.

Table 2 Experimental Results on the KDD CUP dataset

System	*Rank (F1)*	*Rank (precision)*	*Overall precision*	*Overall F1*
Competition best (F1)	1	2*	0.414067	0.444395
Competition best (precision)	2*	1	0.423741	0.426123
Our system, entire article text	10	2	0.387658	0.285263
Our system, Wikilink text	11	3	0.374099	0.284059
Competition mean	18	13	0.254536	0.235321
Competition median	19	15	0.244565	0.232654
Competition worst	37	37	0.050918	0.060285

Note: *This system was not entered for that category, so the rank is extrapolated, not actual.

4.2 TREC 2007

The Text Retrieval Conference (TREC) is organised annually to support research in the field of text retrieval. The conference's workshop is divided into tracks, each of which focuses on a specific application of text retrieval. From 1999 to 2007, the workshop included a QA track, in which participants had to implement and demonstrate systems that could retrieve the answers to each of a set of questions from a large and varied text corpus. A complete technical report on the 2007 QA track and its results can be found in Dang et al. (2007). The scenario for that track was a native English speaker familiar with current events asking 70 series of questions, with each series being composed of approximately ten questions about a different topic relating to people, organisations, events and other news. In total, 51 sets of answers from 21 participants were submitted. We used questions from that final QA track as a test corpus for our system. While this gives us a good test corpus with which to experiment, to the best of our knowledge no one tried using the TREC data for query classification before we did. Therefore, we have no benchmark values with which to compare our results.

Since the competition deals with QA and our system is meant for question-classification, we necessarily had to do two minor modifications to the task in order to make it applicable to our system. First, we ignored the 'other' questions because they were not questions at all but a command to list all other information on the current topic found in the TREC text corpus, and we ignored the second phase of the competition on interactive QA because such an application is outside the current scope of our system. This left us with 445 questions in all 70 series. The second modification was required because the questions in TREC 2007 were asked sequentially, meaning that a system could rely on information from the previous questions, while our system is designed to classify each query by itself with no query history. Consequently, questions that were too vague to be understood without previous information were disambiguated by adding the series' label. For example, the question 'Who is the CEO?' in the series of questions on the company 3M was rephrased as 'Who is the CEO of 3M?' In all cases, we added only the series label, or the strict minimum of information an automated system could easily obtain from the question file's XML data.

We performed two different tests of the classification results we obtained: first on a query basis, and second on a category basis. For the first test, we considered the entire set of categories returned for each query, and we assigned this set into one of the six following classes:

- Completely off-topic: The set of categories has nothing to do with the query. This is the case, e.g., of the query 'Who are members of the board of the International Management Group (IMG)?', which was misclassified into the category 'Northwestern University'.
- Related off-topic: The set of categories has nothing to do with the intended query topic, but is related to a different query with some keywords in common. For example, the query 'On what date was the USS Abraham Lincoln commissioned?' was classified into 'Presidents of the United States', which shows that the system misinterpreted the query as relating to the President of the same name.
- Near-topic: The set of categories is in a wrong subset of the correct topic. For example, this was the case for the query 'How many teenage mutant ninja turtles

were there?', which was classified into the category 'Teenage Mutant Ninja Turtles video games' instead of 'Teenage Mutant Ninja Turtles characters'.

- Generalisation of topic: The set of categories represents high-level general topics, and the intended query topic is a specialisation of them. For example, the query 'What year was the US Mint established?' was classified in the general category 'Mints' instead of the more specific 'United States Mint'.
- Mixed topics: The set of categories contains both on-topic categories and off-topic ones. This was the case of the query 'Into how many languages has Harry Potter and the Goblet of Fire been translated?', which was classified into both the 'Harry Potter in translation' and 'Bhopal' categories.
- On-topic: The set of categories correctly represents the intended query topic. We consider a set of categories on-topic if it consists of some of the same categories as the Wikipedia article of the query's subject matter. For example, the query 'Who was the founder of the Guinness Brewery?' was correctly classified into 'Guinness family'. Moreover, when the article has a category that exactly represents the topic, it must be included in the category set for the classification to be considered on-topic. For example, the category set of any query about the game show Jeopary! had to include the category 'Jeopardy!' in order for us to count it as on-topic.

Those six classes represent the six broad types of classification (or misclassification) our system can do. This test is not a literature standard, but rather aims to give a good picture of how well our system performs, and which mistakes it most commonly does. The results of the test are presented in Table 3. As can be seen from that table, the variant of our system using only wikilink text had some difficulty, and was only in the correct topic for 45% of the queries and on-topic for 24% of them. However, the variant of our system that used the entire article text performed almost twice as well, and was on-topic for 43% of queries and in the correct topic for 73% of them.

Table 3 Assignment of the category set of each TREC query

Variant of our system	*Completely off-topic*	*Related off-topic*	*Near-topic*	*Generalisation of topic*	*Mixed topics*	*On-topic*
Entire article text	52	70	47	64	20	192
Wikilink text	159	86	28	50	17	105

The second test is a literature standard classification precision test, in which we compare our system's classification to the correct classification. This is the same principle that was used to evaluate the system with the KDD CUP queries. We began by manually creating a correct version of the query classification. This was done by using the same criteria as our 'on-topic' class above: each query was classified into the set of categories found in the Wikipedia article of its subject matter. We then computed the precision and F1 value of our system compared to this correct classification in the same way as for the KDD CUP data, using equations (6) to (8). While the principle behind this test is the same as for the KDD CUP evaluation, the test itself is more difficult, because instead of being limited to a set of 67 categories, there are now 300,000 possible categories in which our system could classify each query. The results for each variant of our system are presented in Table 4. These results are consistent with those in Table 3, with the variant of our system that uses the entire article text performing almost twice as well as the variant

using wikilink text only. Compared to Table 2, we find that the wikilink-only variant of our system is doing a bit worse here than in the KDD CUP test, while the entire-text variant is actually doing better than before despite the more challenging classification task.

Table 4 Precision and F1 value of the categories of the TREC queries

Variant of our system	*Precision*	*F1 value*
Entire article text	0.401802	0.351366
Wikilink text	0.220814	0.224213

4.3 Discussion

The results presented in Section 4.1 and Section 4.2 show that our classification system works quite well compared to KDD CUP benchmarks. The wide-ranging and detailed set of categories we used allows it to classify queries in any topic, and can be accurately mapped to a smaller set of categories if needed. Moreover, it was shown that the system can handle both web-style keyword searches and grammatically-correct complete human questions similarly well.

One of the aims of this study was to compare two variants of our system, one built using the entire article text, and the other using only the text found inside the wikilinks of the article. From a theoretical standpoint, the first of these variants would have more data available in each article, and should therefore compute a good result through sheer statistical significance. On the other hand, the wikilinks are the most relevant keywords in each article; so this variant of the system should compute a good result by being limited to a small subset of relevant data. It is hard to predict, from a theoretical standpoint, which of the two variants would produce the best results. The experimental results presented, however, unambiguously show that the variant built using the complete article text is better. This is most visible in the TREC test, where the variant using the entire article text performs nearly twice as well as the variant using wikilink text only. The cause of this result comes from as far back as the R_w weights computed in the first step of the algorithm. These weights represent the significance of each word w in the query, and are computed using the modified tf.idf formula of equation (1). Intuitively, given two words, one of which is seldom used and the other being commonly used, we know that the more significant of the two is the first one, and the result of equation (1) should reflect this intuitive fact. However, the wikilink-only variant of our system fails when dealing with commonly-used words that are only significant in a small set of articles. Such words will occur frequently in articles, and thus, have a low R_w value in the entire-text variant of our system, but they will be in wikilinks only in the small set of articles in which they are actually significant, in accordance with Wikipedia's stylistic guidelines. This false scarcity makes the words appear wrongly significant in the wikilink-only variant of our system. From this false start, the rest of the classification algorithm continues off-track and arrives to the wrong result.

Consider e.g., these three TREC queries: 'Who is the chief executive of the WWE?', 'Who is the chairman of the WWE?', and 'Where is the WWE headquartered?'. All three queries are correctly classified into the 'world wrestling entertainment' category by the entire-text variant of our system, but only the first one is correctly classified by the wikilink-only variant, while the chairman query is misclassified into 'lists of

office-holders' and the headquarters one is sent to 'video game developers'. A human reader would consider *WWE* to be the most significant keyword in all three queries, when compared to *chief*, *executive*, *chairman*, or *headquartered*. Indeed, it is the most seldom-used word and the word with the highest weight in the entire-text variant of our system, as can be seen in Table 5. However, when we consider words in wikilinks only, the occurrence frequency of the four less significant words is greatly reduced: in the most extreme case, headquartered is in wikilinks in less than 6% of the articles in which it appears! This artificial rarity of the words greatly increased their weights in that variant of the system. Meanwhile, *WWE* is in wikilinks in over 77% of the articles in which it appears, and its weight remains practically unchanged. Consequently, as Table 5 shows, *chairman* and *headquartered* appear more significant than *WWE* in the wikilink-only variant of the system.

Table 5 Word statistics for the example queries

Word	W_a, *article text*	W_a, *wikilink text*	R_w, *article text*	R_w, *wikilinks text*
WWE	2,705	2,103	7.8	7.9
Chief	83,977	21,261	5.6	6.1
Executive	82,976	14,025	5.8	6.4
Chairman	40,241	2,704	7.2	8.2
Headquartered	38,749	2,200	7.1	8.0

As we noted, this problem is much more visible in the TREC experiment than in the KDD CUP one. This is because the TREC queries are complete English questions that ask about multiple aspects of a single topic such as the three queries about the WWE we used in the previous example. They therefore rely a lot on less significant words to pinpoint the requested information. On the other hand, the KDD CUP queries are web-style searches where the user only inputs the specific keyword on which he wants information, so they are therefore less likely to include insignificant words that confuse the search algorithm.

As we mentioned in Section 3.2, another difference between the two variants of our system that we studied here appears in step 3 of the search algorithm, when we filter out the title-article pairs in which all query keywords do not occur. The wikilink-only variant has a higher rejection rate than the entire-text variant, but keeps the most significant pairs. We can still wonder if this is a desirable feature. Experimentally, it turns out to be undesirable. Indeed, when ranking the categories in equation (5) of our algorithm, each category receives the sum of weights of the articles pointing to it. It remains true, however, that each article points to several categories (on average seven categories in our database), and each of these categories receives the same weight from that article. Ideally, we would like each query to be classified in as few categories as possible; e.g., the KDD CUP competition rules imposed a maximum of five categories per query. In our algorithm, reducing the number of candidate categories into which a query can be classified is done implicitly by having articles with overlapping category sets. The categories common to several articles are boosted by the summation in equation (5) and surpass categories that appear in only some of those articles. By rejecting too many of the title-article pairs in step 3, the wikilink-only variant hinders the effect of this summation and becomes unable to resolve ties between candidate categories. To illustrate this problem, we present in Table 6 the average number of categories per query for each

variant of our system in each experiment we ran. These results confirm that the variant using wikilink text only classifies the queries into more categories on average. The difference is nearly unnoticeable in the KDD CUP test when using the competition's 67 categories. The reason is that when mapping our system's categories to their more limited set, several different categories are mapped to a single one and collapsed, thus, reducing the average number. To illustrate, Table 6 gives the average statistics both using our categories and after mapping to the competition's 67 categories.

Table 6 Number of categories returned for each variant and each experiment

Variant/experiment	*Average number of categories per query*
Entire article text/KDD CUP, 67 categories	2.04
Wikilink text/KDD CUP, 67 categories	2.15
Entire article text/KDD CUP, our categories	3.56
Wikilink text/KDD CUP, our categories	3.88
Entire article text/TREC	3.74
Wikilink text/TREC	4.97

5 Conclusions

In this paper, we presented a novel approach for query classification using encyclopaedic knowledge mined from the online encyclopaedia Wikipedia. Our method includes both a corpus preparation stage, which can be generalised to any online encyclopaedia, and a statistical classification algorithm that can be applied to any properly prepared corpus. By using Wikipedia in particular, our system gained the ability to classify queries into a set of 300,000 categories covering most of human knowledge and which can easily be mapped to a simpler application-specific set of categories when needed, as well as the ability to recognise and handle uncommon words such as technical terms and typos. The experimental results we presented showed mathematically that our system can handle both complete English questions and web-search-style keyword queries, and that it can classify queries as well as the top classifiers found in practice, as exemplified by the KDD CUP 2005 competition results: our system would have ranked second on precision in that competition, with an improvement of 14% compared to the competition median, and tenth on F1 with a 5% improvement compared to the median. We also considered two variants of our classifier in this study, which differed based on what was considered acceptable article text. We explored the practical impacts of this distinction to determine which of the two variants was best, both for our system and for other systems built on the Wikipedia database.

A lot more information could be mined from Wikipedia to improve our system. We have already discussed in Section 3 the potential benefit of including more semantic information in our corpus, in the form of more descriptive relationships between articles (Völkel et al., 2006) or the ESA between words (Gabrilovich and Markovitch, 2007). Other ideas we are considering include using the hierarchical relationship between the categories generated by the classification algorithm to refine the category weight equation and to filter out irrelevant categories. Alternatively, Ahn et al. (2005) have observed that more relevant words in Wikipedia articles tend to occur earlier in the article

text. We could make use of this observation to refine our word weight equation. Finally, some authors have developed statistical measures of the maturity and quality of Wikipedia articles (Thomas and Sheth, 2007; Lim et al., 2006). Integrating one such metric could make our article weight equation more reliable. These ideas will be explored in future work.

References

Ahn, D., Jijkoun, V., Mishne, G., Müller, K., de Rijke, M. and Sclobach, S. (2005) 'Using Wikipedia at the TREC QA track', *Proceedings TREC 2004*.

Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R. and Ives, Z. (2007) 'DBpedia: a nucleus for a web of open data', *ISWC*, LNCS, Vol. 4825, pp.722–735, Springer.

Beitzel, S.M., Jensen, E.C., Frieder, O, Lewis, D.D., Chowdhury, A. and Kolcz, A. (2005), 'Improving automatic query classification via semi-supervised learning', *Fifth IEEE International Conference on Data Mining*, p.8.

Cucerzan, S. (2007) 'Large-scale named entity disambiguation based on Wikipedia data', *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pp.708–716, Prague.

Dang, H.T., Kelly, D. and Lin, J. (2007) 'Overview of the TREC 2007 question answering track', *Proceedings of the Sixteenth Text Retrieval Conference*.

Fu, J., Xu, J. and Jia, K. (2009) 'Domain ontology based automatic question answering', *International Conference on Computer Engineering and Technology (ICCET '08)*, Vol. 2, Nos. 22–24, pp.346–349.

Gabrilovich, E. and Markovitch, S. (2007) 'Computing semantic relatedness using Wikipedia-based explicit semantic analysis', *Proceedings of the International Joint Conferences on Artificial Intelligence*, pp.1606–1611, Hyderabad, India.

Hu, J., Wang, G., Lochovsky, F., Sun, J-T. and Chen, Z. (2009) 'Understanding user's query intent with Wikipedia', *Proceedings of the 18th International Conference on World Wide Web*, pp.471–480, Spain.

Jansen, M.B.J., Spink, A. and Saracevic, T. (2000) 'Real life, real users, and real needs: a study and analysis of user queries on the web', *Information Processing and Management*, Vol. 36, No. 2, pp.207–227.

Jingbo, Y. and Na, Y. (2008) 'Automatic web query classification using large unlabeled web pages', *Ninth International Conference on Web-Age Information Management (WAIM '08)*, pp.211–215.

Khoury, R. (2009) 'The impact of Wikipedia on scientific research', *Proceedings of the International Conference on Internet Technologies and Applications (ITA09)*, pp.2–11, UK.

Li, Y., Zheng, Z. and Dai, H. (2005) 'KDD CUP-2005 report: facing a great challenge', *ACM SIGKDD Explorations Newsletter*, Vol. 7. No. 2, pp.91–99.

Lim, E-P., Vuong, B-Q., Lauw H.W. and Sun, A. (2006) 'Measuring qualities of articles contributed by online communities', *Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence*, pp.81–87.

Mihalcea, R. (2007) 'Using Wikipedia for automatic word sense disambiguation', *North American Chapter of the Association for Computational Linguistics (NAACL 2007)*, pp.196–203.

Schönhofen, P. (2006) 'Identifying document topics using the Wikipedia category network', *Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence*, pp.456–462.

Shen, D., Pan, R., Sun, J-T., Pan, J.J., Wu, K., Yin, J. and Yang, Q. (2005) 'Q2C@UST: our winning solution to query classification in KDDCUP 2005', *ACM SIGKDD Explorations Newsletter*, Vol. 7, No. 2, pp.100–110.

Thomas, C. and Sheth, A.P. (2007) 'Semantic convergence of Wikipedia articles', *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence*, 2–5 November, pp.600–606.

Viégas, F.B., Wattenberg, M., Kriss, J. and van Ham, F. (2007) 'Talk before you type: coordination in Wikipedia', *Proceedings of the 40th Annual Hawaii International Conference on System Sciences*, p.78, Hawaii.

Völkel, M., Krötzsch, M., Vrandecic, D., Haller, H. and Studer, R. (2006) 'Semantic Wikipedia', *Proceedings of the 15th international conference on World Wide Web*, pp.585–594, Edinburgh, Scotland.

Voss, J. (2005) 'Measuring Wikipedia', *Proceedings of the International Conference of the International Society for Scientometrics and Informetrics*, Sweden.

Wee, L.C. and Hassan, S. (2008) 'Exploiting Wikipedia for directional inferential text similarity', *Proceedings of the Fifth International Conference on Information Technology: New Generations*, pp.686–691.

Notes

1 Available at http://www.wikipedia.org/.

2 Available at http://www.alexa.com/.