Query Classification based on Textual Patterns Mining and Linguistic Features

Mohamed Ettaleb, Chiraz Latiri, Patrice Bellot

Query Classification based on Textual Patterns Mining and Linguistic Features

Mohamed ETTALEB[1], Chiraz LATIRI[2], and Patrice BELLOT[2]

[1] University of Tunis El Manar, Faculty of Sciences of Tunis, LIPAH research Laboratory, Tunis ,Tunisia
[2] Aix-Marseille University, CNRS, LIS UMR 7020, 13397, Marseille, France
mouhamed.taleb@hotmail.fr, chiraz.latiri3@gmail.com, patrice.bellot@lis-lab.fr

Abstract. We argue that verbose natural language queries used for software retrieval contain many terms that follow specific discourse rules, yet hinder retrieval. Through verbose queries, users can express complex or highly specific information needs. However, it is difficult for search engine to deal with this type of queries. Moreover, the emergence of social medias allows users to get opinions, suggestions, or recommendations from other users about complex information needs. In order to increase the understanding of user needs, a tasks, as the CLEF Social Book Search Classification Track, the aims is to investigates how systems can automatically identify book search requests in online forums. In this respect, we introduce in the present paper a new approach to automatically detect the type of each thread. Our proposal aims to identify book search queries by syntactic patterns, association rules between terms and textual sequences mining.

Keywords: Classification · Association Rules · Syntactic Patterns · Sequences Mining.

1 Introduction

The exponential growth of social medias encourage the use of collective intelligence in a spirit of online collaboration. Through these social medias, users can get opinions, suggestions, or recommendations from other members. It is often difficult for users to express their information needs in a search engine query [4]. These social medias allow users to express complex or highly specific information needs through verbose queries. In this context, this type of queries provides detailed information which can not be found in user profile as the expectations of the users and their tastes. Verbose queries, expressed in natural language, are characterized by their length and a complex structure which provide much more context for a more accurate understanding of users needs. Thereby, it is important to find methods that allow to deal effectively with such type of needs. The intrinsic characteristics of this type of query allow us to highlight the use of processes from the NLP to infer semantic aspects and thus potentially better interpret users needs. The Social Book Search Lab on the CLEF 2016 conference

consisted of three tracks: suggestion, mining and interactive track. Within this work we describe our approach on the mining track which is a new one in SBS 2016 edition and investigates two tasks: (*i*) *Classification task*: how Information Retrieval Systems can automatically identify book search requests in online forums, and; (*ii*) *Linking task*: how to detect and link books mentioned in online book discussions.
Our contribution deals only with the classification task. The final objective of this task is to identify which threads on online forums are book search requests. Thereby, given a forum thread with one or more posts, the system should determine whether the opening post contains a request for book suggestions (*i.e.*, binary classification of opening posts). In this respect, we propose to use three types of approaches, namely : an approach based on association rules between terms, textual sequences mining, and an NLP method which relies on nouns, verbs and syntactic patterns extraction (*i.e.*, compound nouns), to improve the classification efficiency. Then, we use the NaiveBayes classifier with WEKA to specify the user's intentions in the requests. In experimental validation, we focus on the analysis of specific needs which are books recommendation but it is worth noting that this approach could be generalized. We believe that specific preprocessing can be set up depending on the nature of the need and the subject of search. The limitations of traditional approaches are overcome, in order to achieve a better representation of user needs. To our knowledge no study has focused on pattern ming combined with NLP technique in this task as source of the query classification.
The paper is structured as follows: related work on classification system is presented in Section 2. Then, a detailed description of our approach is presented in Section 3. Section 4 focuses on an application framework. Section 5 presents the experiments and the results. The Conclusion section wraps up the article and outlines future work.

2 Query classification systems

While considering web search, building a classification system that can automatically classify a large portion of the general query stream with a very good level of accuracy is highly challenging. Web traffic at major search engines often reaches hundreds of millions of queries per day, and web queries are topically ambiguous and very short [10]. Classifying these queries into categories is both a difficult and an important task. So, this problem complicates manual categorization of queries (for studying, or to providing training data) poses great challenges for automated query classification [8]. Many studies in query classification classify queries into many different categories based on the information needed by the user. In [5] considered three different categories of queries, namely: *informational queries*, *navigational queries* and *transactional queries*. The majority of works focus on the type of the queries, rather than topical classification of the queries. Some works are dealing with the problem of classifying queries into a more complex taxonomy containing different topics. To classify queries based on their meaning, some works considered only information available in queries (*e.g.*,

in [2] authors only used terms in queries). Some other works attempted to enrich queries with information from external online datasets, *e.g.*, web pages [5][16] and web directories [17]. The context of a given query can provide useful information to determine its categories. Previous studies have confirmed the importance of search context in query classification. In [3], authors considered a query as a distribution of terms over topics and mapped into high dimensional space for sparse representation, they used LDA topic model for latent topic extraction and word distribution over topics. The Objective of the work is to classify the query into a topic class label by considering the query keywords distributed over various topics. In [15], authors applied feature expansion using word embedding based on word2vec. They applied this approach using GoogleNews and IndoNews data set on Naive Bayes, SVM, and Logistic Regression classifiers. In [12], authors proposed a method which takes advantage of the linguistic information encoded in hierarchical parse trees for query understanding. This was done by extracting a set of syntactic structural features and semantic dependency features from query parse trees. In [7], authors proposed a semi-supervised approach for classifying microblogs into six classes. They initially train a classifier with manually labelled data to probabilistically predict the classes for a large number of unlabelled tweets, then they train a new classifier also using the probabilistically predicted labels for the abovementioned unlabelled tweets, and iterate the process to convergence. The approach introduced in [6] performs query expansion on all query terms upon distinguishing short and verbose queries with promising results. The aims of this method was to show that automatic query classification by verbosity and length is feasible and may improve retrieval effectiveness. Notice that our framework for classifying book search requests in online forums is based on verbose queries, the queries have to be much longer than the conventional ones in a freer style and include a large amount of descriptions of the user's interests.

3 Proposed Approach

The use of verbose queries that have characteristics such as to be long, infers some difficulties to understand user information needs and to interpret them. To overcome this hamper, a subset selection of the original query (*i.e.*, a "subquery") is proposed for improving these queries representation. The impact of the different types of relations that can exist between original query words is focused. For example, some query words may form a compound nouns such as "suggestion book" and others may describe frequent sequences of terms such as "suggest me please". In this section, three methods for book search request classification are proposed.
The first one is based on the sequence mining technique to extract frequent sequences [1] from textual content requests, while the second one consists in extracting linguistic features as verbs, nouns and compound nouns. The last one focused on the association rules to extract useful knowledge and correlations between terms from the queries.

3.1 Textual sequence mining

In order to generate textual frequent sequences features, we propose to extract the frequent and contiguous subsequences in a sequences database and the subsequences whose occurrence frequency is no less than minimum support threshold *minsupp*. In our case, a subsequence is ordered sequence of variable size of k words occurring in the query. We are looking for patterns that are necessarily contiguous, because, if we want to take non−contiguous (gappy) patterns into account, the number of features increases exponentially with the size of the text. Furthermore, most of these patterns will be mere noisy. To overcome both issues, sequential pattern mining can be used to efficiently extract a smaller number of the most frequent features. To address book search requests classification in an efficient and effective manner, we claim that a synergy with some advanced text mining methods in verbose queries, especially sequence mining [1], is particularly appropriate. However, extracting the frequent sequences of words on verbose queries helps to select good features and improve classification accuracy, mostly because of the huge number of potentially interesting frequent sequences that can be drawn from a verbose queries collection (Table 3.1). LCM_seq[3][13] is a variation of LCM[4][18] for sequence mining. It generates all frequently appearing sequence patterns from given database, each transaction is considered to be a sequence. The algorithm follows the scheme so called prefix span, but the data structures and processing method are LCM[18] based.

Definition 1 *sequence* $S = \langle t_1, ..., t_j, ..., t_n \rangle$ *such that* $t_k \in \tau$ *and* n *is its length, is a* $n-termset$ *for which the position of each term in the sentence is maintained.* S *is called a* $n-sequence$.

Most frequent sequences	Support
i_need	18
any_suggestion	13
suggest_me_please	7
any_recommendation_Book	9
any_recommendation_Book_about	5

Table 1. Example of frequent sequences extracted from SBS collection.

3.2 Linguistic features

Textual features extraction is the process of transforming what is essentially a bag of words into a feature set that is usable by a classifier. For the features extraction, two steps are proposed, namely: (1), TreeTagger[5] was used for annotating text with part-of-speech and lemma information. We noticed that the bag of words model is the classical method. It doesn't take into account

[3] http://research.nii.ac.jp/∼uno/code/lcm_seq.html
[4] LCM : Linear time Closed itemset Miner
[5] http://www.cis.uni-muenchen.de/ schmid/tools/TreeTagger/

the order of the words, or how many times a word occurs; (2), only the linguistic features are extracted, so we focus on terms that are either verbs, common nouns and proper nouns. The rationale for this focus is that nouns are the most informative grammatical categories and are most likely to represent the textual content. This selection helps the system to identify the category of queries and enhance accuracy.
Finally, we propose to keep the dependencies and relationships between words through language phenomena. The tagged corpus is used to extract a set of compound nouns by the identification of syntactic patterns as detailed in [9]. The pattern is a syntactic rule on the order of concatenation of grammatical categories which form a noun phrase. 12 syntactic patterns to extract the compound nouns are defined namely:

- 4 syntactic patterns of size two: Noun-Noun, Adjective-Noun, Noun-Verb and Verb-Noun.
- 6 syntactic patterns of size three: Adjective-Noun-Noun, Adjective-Noun-Gerundive, Noun-Verb-Adjective, Verb-Adjective-Noun, Noun-Noun-Verb and Adjective-Noun-Verb.
- 2 syntactic patterns of size four: Noun-Verb-Adjective-Noun and Noun-Noun-Adjective-Noun.

Then, compound nouns are extracted by the identification of syntactic patterns. For example, the compound nouns *suggestion fantasy* and *looking new book* are extracted based on the syntactic patterns respectively Noun-Noun and Verb-Adjective-Noun. The purpose is to extract semantically meaningful unit of text from a query. In the particular task of mining track for classification, units that contain *needs books* are captured. This type of patterns is important because it expresses and contain words that convey opinions about aspects of entities.

3.3 Statistical approach based on association rules inter-terms

The main idea of this approach is to extract a set of non redundant rules, representing inter-terms correlations in a contextual manner. We select these rules that convey the most interesting correlations amongst terms, to help the classifier learn to detect the type of queries.
We generate the association rules using an efficient algorithm for mining all the closed frequent termsets.

Definition 2 *An association rule, i.e., between terms, is an implication of the form* $R : T_1 \Rightarrow T_2$*, where* T_1 *and* T_2 *are subsets of* τ*, where* $\tau := t_1...t_n$ *is a finite set of* n *distinct terms in the collection and* $T_1 \cap T_2 = \theta$*. The termsets* T_1 *and* T_2 *are, respectively, called the premise and the conclusion of* R*. The rule* R *is said to be based on the termset* T *equal to* $T_1 \cup T_2$*. The support of a rule* $R : T_1 \Rightarrow T_2$ *is then defined as:*

$$Supp(R) = Supp(T) \tag{1}$$

while its confidence is computed as:

$$Conf(R) = Supp(T) \div Supp(T1) \tag{2}$$

An association rule R is said to be valid if its confidence value, i.e., $Conf(R)$, is greater than or equal to a user-defined threshold denoted $minconf$. This confidence threshold is used to exclude non valid rules.

3.4 Mining and learning process

The thread classification system serves to identify which threads on online forums are book search requests. Our proposed approach based on text mining is depicted in **Processus 1**. The classification threads process is performed on the following steps: 1) Firstly, annotating the selected threads with part−of−speech and lemma information using TreeTagger. The extracted attributes like nouns, verbs and Nominal syntagms from the annotated threads are denoted by $Feat1$. Secondly, the sequence of terms and the association rules inter-terms are generated using respectively the efficient algorithms **LCM_seq** and **CHARM**. The matrix $M1$(threads_attributes) was built by multiplying five times the positive instances to balance between positive and negative instances. Then, the classification model using the naive Bayesian classifier[6] was generated. Finally, the prediction classification results C_{ti} are Extracted for test requests.

Processus 1 Book search requests classification

Requires: List of recommendation queries file

1: $Q_{Train} \leftarrow < q_1, ..., q_{n-1}, q_n >$
2: $Q_{Test} \leftarrow < q_{T1}, ..., q_{Tn-1}, q_{Tn} >$
3: $Feat_1 \leftarrow$ extract the most frequent sequences of terms and the association rules inter-terms from Q_{Train}
4: Tokenize Q_{Train}
5: Remove stop words from Q_{Train} and Q_{Test}
6: **for each** $q_i \in Q_{Train}$ **do**
7: $Feat_2 \leftarrow$ select syntactic patterns and linguistic features
8: $M_1 \leftarrow Q_{Train}$−features matrix
9: Train the classifier based on these features(*Feat1* and *Feat2*).
10: $M_2 \leftarrow Q_{Test}$−features matrix
11: **for each** $q_{ti} \in Q_{Test}$ **do**
12: Run the classification model.
13: $C_{ti} \leftarrow$ predict class for $Qtest_i$

4 Experimental framework

To evaluate our approach, the data provided by CLEF SBS Mining track track 2016[7] are used. This task focuses on detecting and linking book titles in online book discussion forums, as well as detecting book search requests in forum posts

[6] The Bayesian Classification represents a supervised learning method as well as a statistical method for classification

[7] http://social-book-search.humanities.uva.nl/#/data/mining

for automatic book recommendation. This text mining track is composed of two tasks: classification and linking. The classification task consists to identify book search requests in online forums. The linking task consists to detect and link books mentioned in online book discussions. Concerning the classification task two data sets are used: LibraryThing[8] (LT) and Reddit[9]. These data sets provide discussion threads around several subjects including the recommendation of books provided by users. Note that the linking task use only a data set extracted from LT.

4.1 Reddit

The training data contains threads from the *suggestmeabook subreddit*[10] as positive examples and threads from the books *subreddit* as negative examples. A *subreddit* corresponds to a sub-section of the site devoted to a specific theme, for example, *suggestmeabook subreddit* is a specific sub-section dedicated to book recommendation. In this *subreddit*, it is possible to find queries of this type: "*I'd like to start reading Joyce, but I feel like I need something to break me in to his style. Suggestions?*". On average, the written posts are composed of 5 sentences and 90 words. The Reddit data consists on 248 labelled threads for training, and 89 labelled threads for testing. These posts are expressed in natural language by

```
<?xml version="1.0"?>
<forum type="reddit">
<thread id="2f5rwo">
<category>suggestmeabook</category>
<title>Suggest me a huge, unwieldy tome of a novel.</title>
<posts>
<post id="2f5rwo">
<author>Occurs</author>
<timestamp>1409568526</timestamp>
<parentid></parentid>
<body>Of any genre, are there any long (say, 600 page plus)
 books you'd recommend?</body>
<upvotes>12</upvotes>
<downvotes>0</downvotes>
</post>
</posts>
</thread>
</forum>
```

Fig. 1. Example of Reddit data format

users.

4.2 LibraryThing

LibraryThing is concerning the data set provided for the linking task, 200 threads (3619 posts) labelled with touchstones (links to unique book IDs, on the post

[8] https://www.librarything.com/
[9] https://www.reddit.com/
[10] https://www.reddit.com/r/suggestmeabook

level) for the training are available. In the test, 5097 book titles are identified in 2117 posts. For the classification task, 2000 labelled threads for training and 2000 labelled threads for testing are available. These posts are composed of: the

```
<?xml version="1.0" encoding="UTF-8"?>
<thread id="2/2562">
 <title>Nobel Prize Literature</title>
 <group id="southamericanfictio">South American Fiction-Argentine Writers</group>
 <posts>
   <post id="1">
     <username>carmenDC</username>
     <timestamp>Oct 5, 2006, 6:01pm</timestamp>
     <raw_text>Hello: It's been a while since a Latin American writer won the nobel
     for Literature. Octavio Paz was our last Nobel in 1990.
     Any suggestions on who should be nominated?</raw_text>
     <authors_mentioned/>
     <works_mentioned/>
   </post>
 </posts>
</thread>
```

Fig. 2. Example of LT data format

date the post was written, the content of the post, the post id, the user name and the thread id. For both tasks, the same type of queries is used, an example of book recommendation query: "*I am looking for a non-fiction history book on the background to and the actual time of the Boer War in South Africa. Can anyone suggest any good titles?*". On average, the written posts are composed of 4 sentences and 77 words.

5 Experiments and results

5.1 Classification task experiments

For the evaluation, official measure provides by CLEF SBS 2016, which is accuracy, are used. Four thread sets are used: LibraryThing−TRAIN and Reddit−TRAIN are used for training (TRAIN), LibraryThing−TEST and Reddit−TEST are used for testing (TEST). Then, the feature set of frequent sequences is extracted using LCM_SEQ[11] which is a variation of LCM[12] for sequences mining. LCM is commonly applied to discover frequently appearing patterns. We propose to keep all sequence of words of variable size between 2 and 5. Next, we used CHARM to select all association rules inter-term. As parameters, CHARM takes minsupp = 5 as the relative minimal support and minconf = 0.7 as the minimum confidence of the association rules [11]. While considering the $Zipf$ distribution of each collection, the minimum threshold of the support $minsupp$ values was experimentally set in order to eliminate marginal terms which occur in fewer documents, and therefore they are not statistically important when occurring in a rule. Then, the collection of texts composed of n threads and m features are

[11] http://research.nii.ac.jp/∼uno/code/lcm_seq.html
[12] LCM : Linear time Closed itemset Miner

taken and represents it as an $n \times m$ thread-feature matrix. The elements M_{ij} of the thread-feature matrix are binary and indicate whether a feature exists in the thread or not.

$$M_{i,j} = \begin{cases} 1 & \text{if the feature exists in the thread} \\ 0 & \text{else} \end{cases} \quad (3)$$

Then, using WEKA[13], we analyse the accuracy level (error rate) and compare the performance of the classifiers when using the default setting provided by WEKA. After the training, the test threads are selected. Then, the learned classification model is applied to predict the category of threads, for each of the threads in the test set, the aim is to detect if the opening post of this thread is a book search request or not. We used three learning algorithms, this choice being explained by the fact that they often showed their effectiveness in text analysis tasks: SVM[19], J48[14] and Naive Bayes[20]. Finally, we applied a 10-fold cross validation. In the test sets, eight runs are conducted according to the approaches described in Section 3: four runs on the LIBRARYTHING data collection and four runs on the REDDIT data collection. For each run, we propose to combine different set of features and we used the NaiveBayes classifier that we gave the best scores in the training set(see table5.1)
The following runs are conducted to derive the classification model using Naive

	Naive Bayes classifier		**J48 classifier**		**SVM classifier**	
Detailed Accuracy	Precision	*Accuracy*	Precision	*Accuracy*	Precision	*Accuracy*
Run-NVC	0.844	0.852	0.637	0.639	0.79	0.79
Run-Seq	0.820	0.832	0.630	0.634	0.763	0.771
Run-ART	0.839	0.845	0.626	0.634	0.803	0.810
Run-All-features	0.859	0.869	0.642	0.650	0.812	0.818

Table 2. Comparing classifier algorithms in terms of *Precision* and *Accuracy*

Bayesian classifier:

- **Run-LF(NVC)**: the features Nouns, Verbs and Compound nouns are combined.
- **Run-Seq**: the features of the sequence of words have extracted using LCM_seq algorithm taking *minsupp* =5 as parameters.
- **Run-ART**: We applied the CHARM algorithm to select the set of association rules inter-terms, this latter is combined with the *Run-LF* to extract the classification model.
- **Run-All-features:** the features Compound nouns are combined with the sequence of words using the same parameters as *Run-LF(NV).Seq.*

[13] http://www.cs.waikato.ac.nz/ml/weka/

5.2 Results

The results obtained by our approach on the classification task requests are evaluated in two metrics, which are the *Accuracy* and *Precision*. It simply measures how often the classifier makes the correct prediction. Where: *Accuracy* is the ratio between the number of correct predictions and the total number of predictions (the number of test data points), and *Precision* is the proportion of correct positive classifications from cases that are predicted as positive.

$$\textbf{Accuracy} = (TP + TN) \div (TP + TN + FP + FN) \tag{4}$$

$$\textbf{Precision} = (TP) \div (TP + FP) \tag{5}$$

where: TP is the number of True positives, FP is the number of False positives, TN is the number of True Negatives and FN is the number of False Negative. Table 3 summarizes classification task results conducted on the LibraryThing collection. These results highlight that the combination of Bag of words features (*i.e.* nouns and verbs) and compound nouns performs the best in term of accuracy, (*i.e.*, $Run - LF(NVC)$). We note also that the combination of NLP techniques with patterns mining, (*i.e.*, $Run - All - features$) increases accuracy compared to the use of only NLP techniques. It's noted also that, the runs give very similar results and therefore the differences are not seem really significant. Table 4 describes classification task results conducted on the Reddit collection. We notice also that when we used the sequences of words, the results are well-performed and increase accuracy (*i.e,* $Run - Seq$. However, we noticed that the association rules between terms (*i.e,* $Run - ART$) worked well in the classification of queries, this is justified by the fact that the association rules allowed us to find the terms having a strong correlation with the query's terms. The highest accuracy are obtained in the two collections when we combined all the features(*i.e.*, $Run - All - features$). Finally, all these experiments clearly show that the proposed approach allows to detect significantly the classes of queries. These improvements show the interest of combining the features selection to learning models.

Runs	**Accuracy**	**Precesion**
Run-LF(NVC)	90.98	83.04
Run-Seq	90.24	82,63
Run-ART	90.35	82.75
Run-All-features	**91.13**	**83.18**

Table 3. Classification of the LibraryThing Threads

Runs	**Accuracy**	**Precesion**
Run-LF(NVC)	81.92	74,23
Run-Seq	82.02	74.41
Run-ART	81.75	74.09
Run-All-features	**82.23**	**75.03**

Table 4. Classification of the Reddit posts

Table 5 and 6 present the comparative results with all participants[14]. Our run is best-performed in the evaluation on Reddit collection, and our best results on the LibraryThing collection are ranked sixth in term of accuracy. The best run on Reddit collection is performed with the sequences of words and the verbs as features for classification. This result confirms that mining sequences is useful for classification task. It's worth noting that the obtained results shed light that our proposed approaches, based on text mining and NLP techniques, offer interesting results and helps to identify book search requests in online forums.

Runs	**Accuracy**
Our Best-Run	91.13
Official Run	**94.17**
Medium Run	90.83
Worst Run	74.82

Table 5. Comparison results of the Librarything posts

Runs	**Accuracy**
Our Best-Run	**82.23**
Official Run	82.23
Medium Run	78.65
Worst Run	74.16

Table 6. Comparison results of the Reddit posts

6 CONCLUSIONS

The originality of the proposed approach deals with verbose queries of book recommendation in order to identify the type of request and the associated need. We presented our approach for the 2016 Social Book Search Track, especially for the SBS Mining track. In the different runs dedicated for book search requests classification, we tested four approaches for features selection, namely : Bag of linguistic features (*i.e.*, nouns and verbs), compound nouns, sequences mining and association rules between terms, and their combination. We showed that combining Bag of linguistic features (*i.e.*, nouns and verbs) and compound nouns improves accuracy, and integrating sequences and association rules in classification process enhances the performance. So, the results confirmed that the synergy between the NLP techniques (textual patterns mining and nouns phrases extraction) and the classification system is fruitful.

References

1. Agrawal, R., Srikant, R.: Mining sequential patterns. In: Proceedings of the Eleventh International Conference on Data Engineering. pp. 3–14 (1995)
2. Beitzel, S.M., Jensen, E.C., Frieder, O., Grossman, D.A., Lewis, D.D., Chowdhury, A., Kolcz, A.: Automatic web query classification using labeled and unlabeled training data. In: SIGIR 2005: Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Salvador, Brazil, August 15-19, 2005. pp. 581–582 (2005)
3. Bhattacharya, I., Sil, J.: Query classification using lda topic model and sparse representation based classifier. In: Proceedings of the 3rd IKDD Conference on Data Science, 2016. pp. 24:1–24:2. CODS '16, ACM, New York, NY, USA (2016)

[14] http://social-book-search.humanities.uva.nl//mining16

4. Biskri, I., Rompre, L.: Using associated rules for query reformulation. In: Next Generation Search Engine: Advanced Models for Information Retrieval, pp. 291–303. IGI-Global (2012)
5. Broder, A.Z., Fontoura, M., Gabrilovich, E., Joshi, A., Josifovski, V., Zhang, T.: Robust classification of rare queries using web knowledge. In: SIGIR 2007: Proceedings of the 30th Annual International Conference on Research and Development in Information Retrieval. pp. 231–238 (2007)
6. Buccio, E.D., Melucci, M., Moro, F.: Detecting verbose queries and improving information retrieval. Inf. Process. Manage. **50**(2), 342–360 (2014). https://doi.org/10.1016/j.ipm.2013.09.003, https://doi.org/10.1016/j.ipm.2013.09.003
7. Chen, Y., Li, Z., Nie, L., Hu, X., Wang, X., Chua, T., Zhang, X.: A semi-supervised bayesian network model for microblog topic classification. In: COLING 2012, 24th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, 8-15 December 2012, Mumbai, India. pp. 561–576 (2012)
8. Gravano, L., Hatzivassiloglou, V., Lichtenstein, R.: Categorizing web queries according to geographical locality. In: Proceedings of the 2003 International Conference on Information and Knowledge Management. pp. 325–333. CIKM (2003)
9. Haddad, H.: French noun phrase indexing and mining for an information retrieval system. In: String Processing and Information Retrieval, 10th International Symposium. pp. 277–286 (2003)
10. Jansen, B.J., Spink, A., Saracevic, T.: Real life, real users, and real needs: a study and analysis of user queries on the web. Inf. Process. Manage. **36**(2), 207–227 (2000)
11. Latiri, C.C., Haddad, H., Hamrouni, T.: Towards an effective automatic query expansion process using an association rule mining approach. J. Intell. Inf. Syst. **39**(1), 209–247 (2012)
12. Liu, J., Pasupat, P., Wang, Y., Cyphers, S., Glass, J.R.: Query understanding enhanced by hierarchical parsing structures. In: 2013 IEEE Workshop on Automatic Speech Recognition and Understanding, Olomouc, Czech Republic, December 8-12, 2013. pp. 72–77 (2013)
13. Nakahara, T., Uno, T., Yada, K.: Extracting promising sequential patterns from RFID data using the LCM sequence. In: Knowledge-Based and Intelligent Information and Engineering Systems - 14th International Conference, KES 2010, Cardiff, UK, September 8-10, 2010, Proceedings, Part III. pp. 244–253 (2010)
14. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann (1993)
15. Setiawan, E.B., Widyantoro, D.H., Surendro, K.: Feature expansion using word embedding for tweet topic classification. 2016 10th International Conference on Telecommunication Systems Services and Applications (TSSA) pp. 1–5 (2016)
16. Shen, D., Pan, R., Sun, J., Pan, J.J., Wu, K., Yin, J., Yang, Q.: Query enrichment for web-query classification. ACM Trans. Inf. Syst. **24**(3), 320–352 (2006)
17. Shen, D., Sun, J., Yang, Q., Chen, Z.: Building bridges for web query classification. In: SIGIR 2006: Proceedings of the 29th Annual International Conference on Research and Development in Information Retrieval. pp. 131–138 (2006)
18. Uno, T., Kiyomi, M., Arimura, H.: Lcm ver.3: Collaboration of array, bitmap and prefix tree for frequent itemset mining. In: Proceedings of the 1st International Workshop on Open Source Data Mining: Frequent Pattern Mining Implementations. pp. 77–86. OSDM '05, ACM, New York, NY, USA (2005)
19. Vosecky, J., Leung, K.W.T., Ng, W.: Searching for quality microblog posts: Filtering and ranking based on content analysis and implicit links. In: Lee, S.g., Peng, Z.,

Zhou, X., Moon, Y.S., Unland, R., Yoo, J. (eds.) Database Systems for Advanced Applications. pp. 397–413. Springer Berlin Heidelberg, Berlin, Heidelberg (2012)
20. Yuan, Q., Cong, G., Thalmann, N.M.: Enhancing naive bayes with various smoothing methods for short text classification. In: Proceedings of the 21st International Conference on World Wide Web. pp. 645–646. WWW '12 Companion, ACM, New York, NY, USA (2012). https://doi.org/10.1145/2187980.2188169, http://doi.acm.org/10.1145/2187980.2188169