IRIT-QFR: IRIT Query Feature Resource

Serge Molina [(1)], Josiane Mothe [(1)], Dorian Roques [(1)],
Ludovic Tanguy [(2)], and Md Zia Ullah [(1)]

(1) IRIT, UMR5505, CNRS & Univ. Toulouse, France,
(2) CLLE-ERSS, UMR 5263, CNRS & Univ. Toulouse, France

Abstract. In this paper, we present a resource that consists of query features associated with TREC adhoc collections. We developed two types of query features: linguistics features that can be calculated from the query itself, prior to any search although some are collection-dependent and post-retrieval features that imply the query has been evaluated over the target collection. This paper presents the two types of features that we have estimated as well as their variants, and the resource produced. The total number of features with their variants that we have estimated is 258 where the number of pre-retrieval and post-retrieval features are 81 and 171, respectively. We also present the first analysis of this data that shows that some features are more relevant than others in IR applications. Finally, we present a few applications in which these resources could be used although the idea of making them available is to foster new usages for IR.

Keywords: Information systems, Information retrieval, Query features; IR resource; Query feature analysis

1 Introduction

Query features are features that can be associated with any query. They have been used in information retrieval (IR) literature for (1) query difficulty prediction and (2) selective query expansion; however, they can be useful for other applications.

In this paper, we present a resource that we have developed, which associates features to queries considering several TREC collections and which considers many different approaches: linguistic versus statistic-based, pre- and post-retrieval, and collection-dependent and -independent. This resource is to be made available to the IR community.

In the literature of query difficulty prediction, query features are categorized into two groups, according to the fact that the feature can be calculated prior any search (pre-retrieval feature) or not (post-retrieval feature) [2]. An example of a pre-retrieval feature is IDF_Max which is calculated as the maximum of the IDF term weight (as computed when indexing the document collection) over the query terms. High IDF means the term is not very frequent, thus high IDF_Max for a query means that this query contains at least one non-frequent term. On

the other hand, an example of a post-retrieval feature is NQC (Normalized query commitment), which is based on the standard deviation of the retrieved document scores [13]. A high standard deviation means that the retrieved documents obtained very different scores meaning the retrieved document set is not homogeneous.

As for pre-retrieval features, we can also make a distinction between features that can be calculated independently to any document collection and the ones that need the document collection in some way (obviously, post-retrieval features are collection dependent since they are calculated over a retrieved document set). Going back to IDF_Max, it is obviously dependent on the document collection. On the other hand, SynSet (the average number of senses per query term as extracted from WordNet) [10] is collection-independent, since it only requires access to the query terms in order to be calculated.

In our work, we extract both pre- and post-retrieval features. We also distinguish between collection dependent and collection independent features. The details of the feature definitions can be found in Section 2 as well as the collections on which the features are already available.

In Section 3, we provide the first analysis of the data. In Section 4, we introduce a few applications that make use of such features. Finally, Section 5 concludes this paper.

2 Query Features

In this section, we describe the query features that we have estimated as well as their variants, including pre- and post-retrieval features.

2.1 Document collection independent pre-retrieval features: WordNet-based and Other linguistic features

WordNet-based features (pre-retrieval) WordNet-based features are pre-retrieval and document collection independent.

WordNet is a linguistic resource that interlinks senses of words (represented as sets of synonyms, or synsets) and labels the semantic relations between word senses [9]. The original (Princeton) version contains more than $117,000$ synsets and more than $150,000$ unique entries (source: `https://wordnet.princeton.edu/` on the 5th of March 2017).

Figure 1 presents an extract of WordNet for the term "tiger." WordNet distinguishes different relationships between terms as follows:

1. Synonyms: words that denote the same concept and are interchangeable in many contexts. Synonyms are terms that belong to the same Synset, such as "tiger" and "panthera tigris."
2. Hyponyms/Hypernyms: these relationships link more generic synsets to specific ones. While "Panthera tigris" is a *hypernym* of "Bengal tiger"; the latter is a *hyponym* of the former.

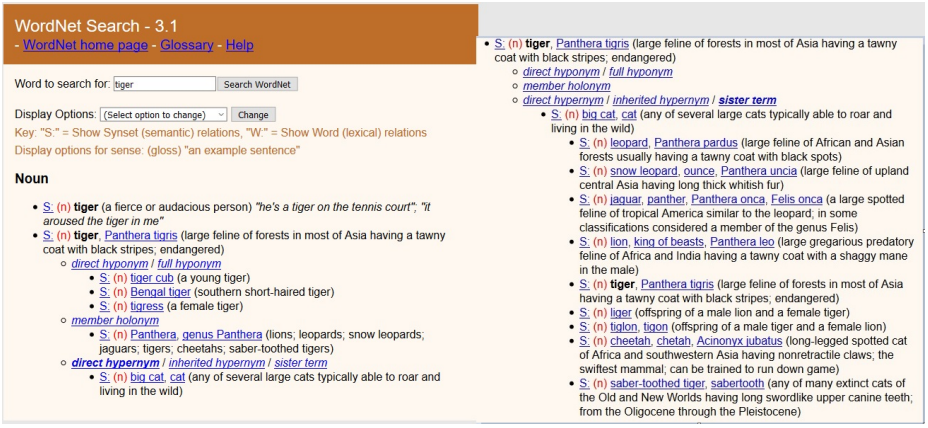

Fig. 1. Extract of WordNet for the term **tiger**. The right-side part of the figure provides the sister terms from the term tiger while the left-side part provides the two senses of the word and details for the second sense as well as direct hyponyms, holonym, and direct hypernym.

3. Meronym/Holonyms: correspond to the part-whole relation. Y is a *meronym* of X means Y is a part of X; in that case, X is a *holonym* of Y.
4. Sister-terms: *Sister-terms* are terms that share the same hypernym.

We use this resource as follows: for each query term, we count the associated terms of each type (e.g. number of sister terms for each query term) and then aggregate the obtained values for a given relationship over query terms for a given query. Since a query may contain several query terms, given a query and a relationship type (e.g. synonym or sister-term), we calculate the following aggregations to get a single value of the feature variant for each query:

- minimum, maximum, mean, and total: the minimum (maximum, average, and total) number of the terms associated with the query terms when using the relationship, over the query terms;
- Q1, median, Q3: for each query term, we calculate the number of associated terms using the relationship. These numbers are first sorted in increasing order; then the set is divided into quartiles. Q1 (median, Q3) is the value that makes at least one quarter (2 quarters, 3 quarters) of the numbers having a score lower than Q1 (median, Q3).
- Standard deviation (std) and variance (or std^2): standard deviation refers to the square root of the mean of the squared deviations of the number of terms associated with the query terms from their mean and variance is the squared value of the standard deviation.

Other linguistic features (pre-retrieval) We also consider the linguistic features as defined by Mothe and Tanguy [10]. These are also pre-retrieval features

and collection independent. The queries have been analysed using generic techniques (POS tagging and parsing) from the Stanford CoreNLP suite [8]. We have also used the CELEX morphological database [1] for assessing the morphological aspects of the query terms.

All the features are computed without any human interaction, and as such are prone to processing errors. The 13 linguistic features are presented in Table 1[1], categorized according to their level of linguistic analysis. Calculation details can be found in [10].

Table 1. Linguistic features as defined in [10].

Feature Name	Description
	Lexical features
NBWORDS	Number of words (terms) in the query
LENGTH	Word length in number of characters
MORPH	Average number of morphemes per word (according to CELEX)
SUFFIX	Number of suffixed words (based on suffixes extracted from CELEX)
PN	Number of proper nouns (according to CoreNLP's POS tagger)
ACRO	Number of acronyms
NUM	Number of numeral values (dates, quantities, etc.)
UNKNOWN	Number of unknown tokens (based on WordNet)
	Syntactical features
CONJ	Number of conjunctions (according to CoreNLP's POS tagger)
PREP	Number of prepositions (idem)
PP	Number of personal pronouns (idem)
SYNTDEPTH	Syntactic depth (maximum depth of the syntactic tree, according to CoreNLP parser)
SYNTDIST	Syntactic links span (average distance between words linked by a dependency syntactic relation)

2.2 Document collection dependent pre-retrieval features

Finally, as pre-retrieval features, we also consider IDF (Inverse document frequency), which is extracted from the document indexing file (we use LEMUR index for that since it gives a direct access to it). Moreover, IDF statistics across IR tools are consistently used in previous research [5]. As opposed to the other pre-retrieval features presented upper, IDF is collection dependent. We calculate the same 9 variants as previously: minimum, maximum, mean, total (sum), Q1, median, Q3, standard deviation, and variance of term-IDF score over the query terms.

[1] [10] paper presents the SynSet feature which is one of the features based on WordNet and thus included in Section 2.1.

2.3 Post-retrieval features

Post-retrieval features are by definition collection dependent. These features are extracted either from the query-document pairs or retrieved documents.

Letor-based post-retrieval features Letor features have been used in learning to rank applications [12]. In Letor, these features are associated with query-document pairs. For example, *BM25.0* corresponds to the score as obtained using BM25 model for a given query; it is thus attached to a query-document pair. We use Terrier platform[2] to calculate the Letor features. The Terrier platform has implemented the Fat component, which allows to compute many features in a single run [7]. More details on the Letor features can be found on Letor collection description[3]. One of them is PageRank which can be calculated for linked documents only (for this reason this feature cannot be calculated for the TREC Robust collection).

In the feature names (Table 2), SFM stands for SingleFieldModel and means that the value corresponds to score, which a document obtained using the mentioned search model (LM stands for Language Model, DIR for Dirichlet smoothing and JM for Jelinek-Mercer smoothing). A .0 means that the calculations have been made on the document's title only; while .1 means they have used the entire document content. The features that do not contain SFM are measures calculated from the occurrences of query terms in the retrieved documents (e.g. mean_tf is the mean of TF (term frequency) of query terms in the considered document).

To make the Letor features usable as query features, we have aggregated them over the retrieved documents for a given query. For example, we calculate the mean of the BM25 scores over the retrieved document list for the considered query. We have used the same 9 aggregation functions as presented in Section 2.1 (minimum, maximum, mean, total (*Nbdoc*), Q1, median, Q3, standard deviation, and variance). Nbdoc is not an aggregation value since it corresponds to the number of documents retrieved for the given query given the retrieval model used.

PageRank features We also calculate two PageRank features: PageRank_prior and PageRank_rank, when dealing with linked documents, that means, for WT10G and GOV2 collections. We use Lemur implementation of the PageRank feature. To generate the variants of the PageRank features, we have used the same 9 aggregation functions as previously over the retrieved documents for a query.

2.4 Collections for which the features have been estimated

In total, we calculated 258 individual features. So far, these features have been calculated on three TREC data collections from the adhoc task: Robust, WT10G, and GOV2.

[2] `http://terrier.org/docs/v4.0/learning.html`
[3] `https://www.microsoft.com/en-us/research/project/mslr/`

Table 2. Post-retrieval features as defined for Letor in [12] and [7]

Feature Name	Description
	Calculated using Terrier module
WMODEL.SFM.Tf.0 and .1	The value of the TF score for the query and the document title/body
WMODEL.SFM.TF_IDF.0 and .1	The value of the TF*IDF score for the query and the document title/body
WMODEL.SFM.BM25.0 and .1	The value of BM25 score for the query and the document title/body
WMODEL.SFM.DirLM.0 and .1	The score value for the language model with Dirichlet smoothing for the query and the document title/body
QI.SFM.Dl.0 and .1	Number of terms in the document title/body
Dirichlet.Mu1000	The score value for the language model with Dirichlet smoothing with the smoothing parameter = 1000, for the whole document
JM.col.λ.0.4doc.λ0.0	The score value for the language model with Jelinek-Mercer smoothing, with a collection lambda of 0.4
	Calculated using Lemur
sum_tf_idf_full	The sum of TF*IDF values for the query terms
mean_tf_idf_full	The mean of TF*IDF values for the query terms
sum_tf_full	The sum of TF values for the query terms
mean_tf_full	The mean of TF values for the query terms
pagerank_rank	The rank of document based on PageRank scores
pagerank_prior	The log probability of PageRank scores

For Robust collection[4], TREC competition provided approximately 2 gigabytes of newspaper articles including the Financial Times, the Federal Register, the Foreign Broadcast Information Service, and the LA Times [14]. The TREC WT10G collection is composed of approximately 10 gigabytes of Web/Blog page documents [6]. The GOV2 collection includes 25 million web pages, which is a crawl of .gov domain [3].

The three test collections consist also of topics that comprise a topic title which we use as the query. There are 250 topics in the Robust collection, 100 topics in the WT10G collection, and 150 topics in the GOV2 collection.

Now that we have either implemented new code or gathered codes to estimate the query feature values, there is no limit to calculate the features for other collections; that we plan to do in the next months. ClueWeb09B and Clueweb12B are the short term targets. Moreover, we will continue to gather new query features.

[4] `http://trec.nist.gov/data/robust.html`

We make available the feature resource to foster new usages for IR, the resource is available to download at `http:/doi.org/10.5281/zenodo.815319` (proper user agreements). If you use this resource in your research, it is required to cite the following paper:

S. Molina, J. Mothe, D. Roques, L. Tanguy, and M. Z. Ullah. IRIT QFR: IRIT Query Feature Resource. In Experimental IR Meets Multilinguality, Multimodality, and Interaction 8th International Conference of the CLEF Association, CLEF2017, Dublin, Ireland, September 11-14, 2017, Proceedings, volume 10439, 2017.

3 Analysis of the resource

In this section, we provide some elements of the descriptive analysis of the resources we have built and presented in the previous sections.

3.1 Descriptive analysis

In order to have an idea of the trends of feature variants, in Figure 2, we show the boxplots associated to 4 query features and their variants. In a given boxplot, each query makes a contribution. For the 2 linguistic features, we did not plot the variance since the high values would have flattened the others. For the 2 post-retrieval features, we did not plot the number of retrieved documents for the same reason.

Since we can assume that the queries are diverse in many senses in the TREC collections (e.g. in terms of difficulty, in terms of specificity, ...), one interesting insight can be to know how much the different features and variants vary according to the queries.

On Figure 2, we can see that the Synonyms and Sister terms features (which are calculated on the query only) have the same trends when considering their different variants.

When considering the BM25.0 (calculated on document titles only), we can see on Figure 2 that most of the queries got a null value for the min, Q1, median, and Q3 variants. The feature variant that varies the most is the variance and in a little smaller extend the max variant. The null value for the min, Q1, and median variants holds for all the features calculated on the title but one that got negative value which is the Jelinek.Mercer.collectionLambda0.4.documentLambda0.0 feature (see Figure 3).

When considering the BM25.1 (calculated on the entire documents), we can see on Figure 2 that the values are higher than when calculated on the title only (which is indeed an expected result), and that the null phenomenon does not hold on. Still, the variance is the variant that varies the most, however, max and min values are also on a quite large scale.

Figure 3 presents the median variant for several Letor features. On the left side part of the figure, which represents the values when the title of the documents is considered, we can see that the values are not at the same scale and that the Jelinek-Mercer smoothing is i) the one that varies the most and ii)

the only one that is negative. On the right side of the figure, we removed the Jelinek-Mercer values since they would have hidden the other values variation. We can see as for BM25 in Figure 3 that the values vary more when considering the entire document than when considering the title only.

The NbDoc (Number of documents retrieved) variant is also somehow interesting: while it is often equal to its maximum value $1,000$ (this value comes from the way we configured Terrier when calculating the features), for a few queries, its value is lower. Figure 4 displays the values for a few models.

In the various previous figures, we display the results for WT10G; but the same type of conclusions can be made using Robust and GOV2.

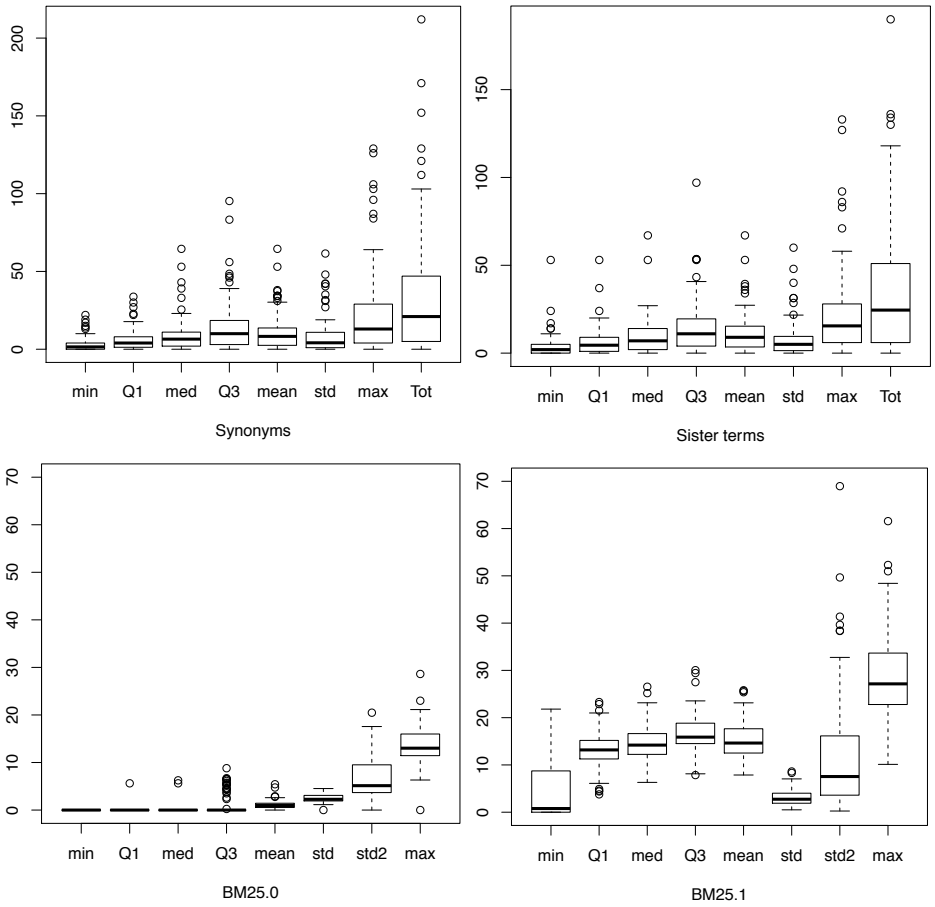

Fig. 2. Boxplots of the Synonyms, Sister_Termes, BM25.0, and BM25.1 features and their variants - WT10G Collection.

4 Applications

One possible application as mentioned previously in this paper is query difficulty prediction. Figure 5 displays the plots of NDCG (Y-axis) as calculated from a BM25 run using default parameters in Terrier and four of the query features (X-axis) using WT10G collection.

Alternatively, Pearson correlation can be calculated in order to measure the link between a single feature and actual system effectiveness. More concretely, we

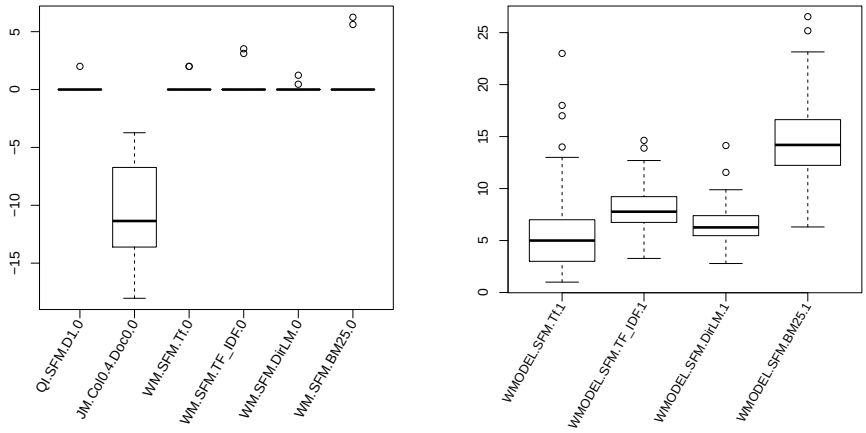

Fig. 3. Boxplots of the median variant for various features when calculated on title only (0) and on the entire document (1) - WT10G Collection. For the variants calculated on the title only, let us mention that the null values hold for min and Q1 as well.

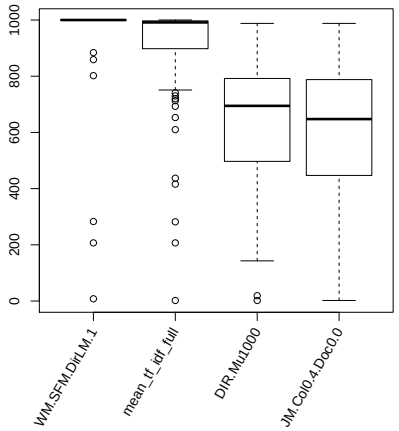

Fig. 4. Boxplots of the NbDoc (number of retrieved documents) variant for a few features from Letor features - WT10G Collection.

investigate the combination of query predictors in order to enhance prediction. As for query predictors, we consider the features presented in this paper.

We developed another application in [4], where query features are used in a machine learning model based on learning to rank principle in order to learn which system configuration among a variety of configurations should be used to best treat a given query. The candidate space is formed of tens of thousands of possible system configurations, each of which sets a specific value for each of the system parameters. The learning to rank model is trained to rank them with respect to an IR performance measure (such as nDCG@1), thus emphasizing the importance of ranking "good" system configurations higher in the ranked list.

Moreover, the approach makes a query-dependent choice of system configuration, i.e. different search strategies could be selected for different types of a query; based on query features. In that study, a subset of the features we present

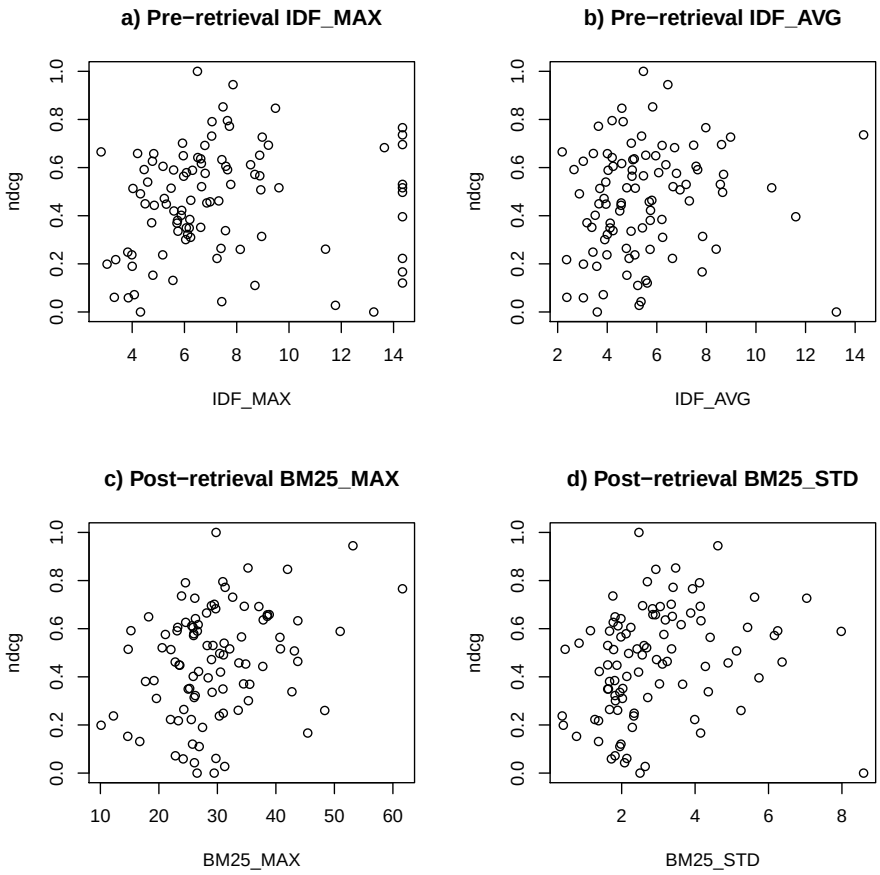

Fig. 5. Plots of NDCG (Y-axis) and query features (X-axis) - WT10G Collection.

in this paper have been used (linguistic features from [10] and IDF variants). The paper shows that this approach is feasible and significantly outperforms the grid search way to configure a system, as well as the top performing systems of the TREC tracks.

In current work, we are developing a new method that aims like in [4], at optimizing the system configuration on a per-query basis [11]. Our method learns the configuration models in a training phase and then explores the system feature space and decides what should be the system configuration for any new query. The experiments on TREC 7 & 8 topics from adhoc task show that the method is very reliable with good accuracy to predict a system configuration for an unseen query. We considered about 80,000 different system configurations.

Figure 6 shows a comparison between the method we proposed in [11] (red square line) and the weak baseline (the configuration that provides the best results in average over the queries) in one hand and the ground truth on the other hand (when the best system is used for each query), represented by the blue and purple straight lines respectively. We also compare the results with a fair baseline that corresponds to a method that learns on a limited predefined set of systems only (green triangles). Figure 6 reports the MAP over the set of unseen test queries (averaged over the 10 draws resulting from 10-folds cross-validation) using the predicted configuration.

5 Conclusion

This paper presents a new resource that associates many features to queries from TREC collections. We distinguish between pre- and post-retrieval features as well as between collection-dependent and collection-independent features. Some

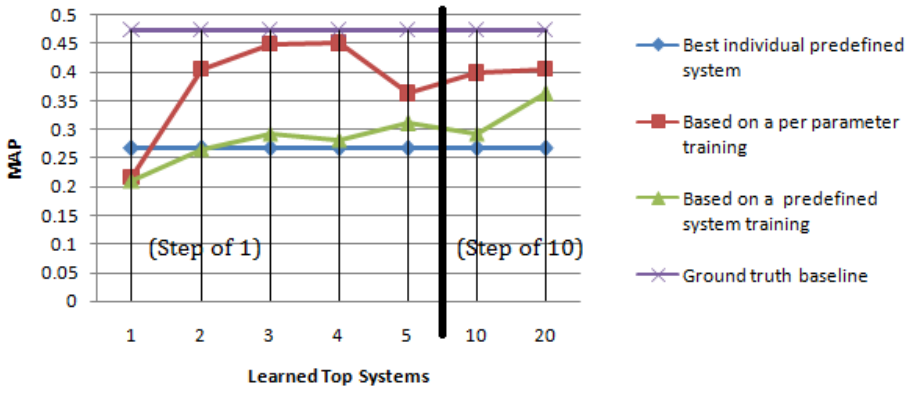

Fig. 6. MAP - A comparison between the best predefined system, the PPL method, the predefined system classifier, and the ground truth method. The predictive functions are trained on top 1, 2 ,3, 4, 5, 10, and 20 systems per query (X-axis).

features have linguistic basis while others are based on statistics only. We use features from the literature but also new features that are generated from both WordNet linguistic resource and Letor learning to rank document-query pairs features.

We have already used some of these features in applications related to system configuration selection, query difficulty prediction, and selective query expansion, but we think these resources could also be used for other applications.

In our future work, we aim at developing this resource for other collections. We are targeting ClueWeb09B and ClueWeb12B, but also other collections such as TREC Microblog-based collections.

References

1. R. H. Baayen, R. Piepenbrock, and L. Gulikers. The celex lexical database (release 2). *Linguistic Data Consortium, Philadelphia*, 1995.
2. D. Carmel and E. Yom-Tov. Estimating the query difficulty for information retrieval. *Synthesis Lectures on Information Concepts, Retrieval, and Services*, 2(1):1–89, 2010.
3. C. L. Clarke, N. Craswell, and I. Soboroff. Overview of the trec 2004 terabyte track. In *TREC*, volume 4, page 74, 2004.
4. R. Deveaud, J. Mothe, and J.-Y. Nie. Learning to rank system configurations. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pages 2001–2004. ACM, 2016.
5. C. Hauff, V. Murdock, and R. Baeza-Yates. Improved query difficulty prediction for the web. In *Proceedings of the 17th ACM conference on Information and knowledge management*, pages 439–448. ACM, 2008.
6. D. Hawking. Overview of the trec-9 web track. In *TREC*, 2000.
7. C. Macdonald, R. L. Santos, I. Ounis, and B. He. About learning models with multiple query-dependent features. *ACM Transactions on Information Systems (TOIS)*, 31(3):11, 2013.
8. C. D. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. J. Bethard, and D. McClosky. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60, 2014.
9. G. A. Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.
10. J. Mothe and L. Tanguy. Linguistic features to predict query difficulty. In *ACM Conference on research and Development in Information Retrieval, SIGIR, Predicting query difficulty-methods and applications workshop*, pages 7–10, 2005.
11. J. Mothe and M. Washha. Predicting the best system parameter configuration: the (per parameter learning) ppl method. In *21st International Conference on Knowledge-Based and Intelligent Information & Engineering Systems*, 2017.
12. T. Qin, T.-Y. Liu, J. Xu, and H. Li. Letor: A benchmark collection for research on learning to rank for information retrieval. *Information Retrieval*, 13(4):346–374, 2010.
13. A. Shtok, O. Kurland, and D. Carmel. Predicting query performance by query-drift estimation. In *Conference on the Theory of Information Retrieval*, pages 305–312. Springer, 2009.
14. E. M. Voorhees. The trec robust retrieval track. In *ACM SIGIR Forum*, volume 39, pages 11–20. ACM, 2005.