A selective approach to stemming for minimizing the risk of failure in information retrieval systems

Gökhan Göksel[1], Ahmet Arslan[1], and Bekir Taner Dinçer[2]

[1]**Department of Computer Engineering, Eskişehir Technical University,Eskişehir, Turkey**
[2]**Department of Computer Engineering, Muğla Sıtkı Koçman University, Muğla, Turkey**

Corresponding author:
Gökhan Göksel[1]

Email address: gciplak@eskisehir.edu.tr

ABSTRACT

Stemming is supposed to improve the average performance of an information retrieval system, but in practice, past experimental results show that this is not always the case. In this paper, we propose a selective approach to stemming that decides whether stemming should be applied or not on a query basis. Our method aims at minimizing the risk of failure caused by stemming in retrieving semantically related documents. The proposed work mainly contributes to the IR literature by proposing an application of selective stemming and a set of new features that derived from the term frequency distributions of the systems in selection. The method based on the approach leverages both some of the query performance predictors and the derived features and a machine learning technique. It is comprehensively evaluated using three rule-based stemmers and eight query sets corresponding to four document collections from the standard TREC and NTCIR datasets. The collections, except one of them, include Web documents ranging from about 25 million to 733 million. The results of the experiments show that the method can make accurate selections that alleviate the per query performance losses. Also, the method systematically improves the average retrieval performance of the single systems in the selection for most of the query sets.

Keywords:Selective information retrieval, Selective stemming, Robustness

INTRODUCTION

Stemming serves two purposes in the context of Information Retrieval (IR): *i*) reducing the index size and *ii*) recall enhancement in the document lists retrieved for queries. Mapping actual query terms to their base forms reduces the amount of unique terms in any given collection and as a result it proportionally reduces the size of the index. In this respect, stemming serves well and there is no counter factual issue reported in the IR literature. On the other hand, this process also increases the relative frequency of terms within a document by unifying morphological variants of the terms and hence it is supposed that it increases the level of recall in the document lists retrieved by IR systems to any given query. However, this unification process may unify morphologically related but semantically unrelated two terms into a single base form. In such a case, stemming causes those documents which are semantically unrelated to a given query to become, unexpectedly, a part of the document list retrieved for that query. That's why stemming harms the performance for some queries by reducing the level of recall in the document lists retrieved for the queries.

It has been a long-standing debate whether stemming contributes to the effectiveness of an IR system (Harman, 1991; Krovetz, 1993; Alotaibi and Gupta, 2018). On this account, in the work of Harman (1991), it is shown that stemming does not affect average retrieval performance. Harman says that stemming affects both positively and negatively to an almost equal number of queries and hence the average remains the same. However, in contrast, the works of Krovetz (1993) and Hull (1996) provide evidence that counts against those negative empirical findings and relate them to the system and the query sets examined in the

45 work of Harman. In this respect, it can be said that stemming may or may not contribute to the retrieval
46 effectiveness of an IR system depending on the query that the system responds to.

47 In this paper, to decide whether stemming should be applied to any given query, a selective approach
48 is proposed. The approach is based on k-Nearest Neighbors binary classifier and it employs a base set
49 of existing pre-retrieval query performance features from the IR literature, including inverse document
50 frequency (IDF), query scope (He and Ounis, 2004a), and average similarity between collection and query
51 (AvgSCQ) (Zhao et al., 2008). In addition, a set of new features that are derived from the frequency
52 distributions of query terms is introduced in this paper and used for the binary classifier (Sect. Proposed
53 Selective Approach to Stemming). The proposed selective approach follows the common IR practice,
54 where BM25 (Robertson and Zaragoza, 2009) is used as the term weighting model and rule-based
55 stemmers.

56 It is expected that an IR system fulfill every information need of users at an acceptable level of
57 satisfaction. Such a system refers to a robust system that evenly distributes its total (average) effectiveness
58 on every query. If a system applies stemming and its average performance remains unchanged (the results
59 of the work of Harman), it is highly likely that the system increases performance for some queries and
60 decreases performance for some other queries: that is, the system diverges from being robust. Such
61 situations lead to high variation in performance across queries, and hence harms the robustness of IR
62 systems. The proposed selective approach is a remedy to this problem and it is capable of providing
63 robustness in retrieval effectiveness for the IR systems employing stemming (Sect. Evaluation of the
64 Selective Approach to Stemming). By alleviating the problem, the performance of an IR system can be
65 improved more than the expected performance of the system in which stemming is naively applied. In
66 this perspective, the paper makes a contribution to the IR literature by considering the importance of the
67 selective application of stemming in IR systems.

68 In particular, we address the following research questions:

69 *RQ1* To what degree the proposed selective approach is accurate in predicting the queries that stemming
70 should (not) be applied to.

71 *RQ2* Does the proposed selective approach contribute to the robustness in retrieval effectiveness of the
72 IR systems that employ stemming?

73 In summary, the work presented in this paper contributes to the literature by proposing a selective
74 approach to stemming. The work aims at minimizing the risk of failure in retrieving semantically related
75 documents to the employed stemming algorithm for any given query. We have achieved this aim by
76 accurately predicting whether stemming should be applied to a given query using a machine-learning
77 technique. We have used a set of query performance predictors from the literature and new features for
78 this purpose. Also, the risk-sensitive analysis shows that the proposed work increases the robustness of
79 an IR system by minimizing the performance fluctuation across queries caused by a failure in applying
80 stemming to the IR system. According to the experimental results, the proposed selective method not
81 only increases the overall retrieval performance in most cases but also contributes to the robustness of the
82 IR system together. The selective method validates its contribution to the robustness of the IR system
83 according to the performed experiments on diverse sets of queries and generalizes to the rule-based
84 stemming algorithms. Considering the contribution, aim, and used method together, the proposed paper
85 includes pioneering work to our best knowledge.

86 The rest of the paper is organized as follows. The motivation of this work is provided in Sect.
87 Motivation. The section Related Works reviews the related works about selective IR, stemming, and
88 selective stemming. Differences between the proposed work from prior work are also presented in the
89 section. Details of the proposed work are given in Sect. Proposed Selective Approach to Stemming.
90 The experimental evaluations are performed on a wide range of standard set of queries from TREC and
91 NTCIR (Sect. Experimental Setup). The used document collections of the corresponding query sets
92 are based on Web corpora that includes ClueWeb09-B, ClueWeb12-B13, GOV2, and, also Wall Street
93 Journal (WSJ) newspaper articles in TIPSTER collection is used. On the query sets, the evaluation results
94 (Sect. Evaluation of the Selective Approach to Stemming) show that our selective method is on average
95 more effective and robust than the considered single systems that are participated in selection procedure
96 for most of the query sets and stemming algorithms. Implications of this work are discussed in Sect.
97 Implications from theoretical and practical perspective.

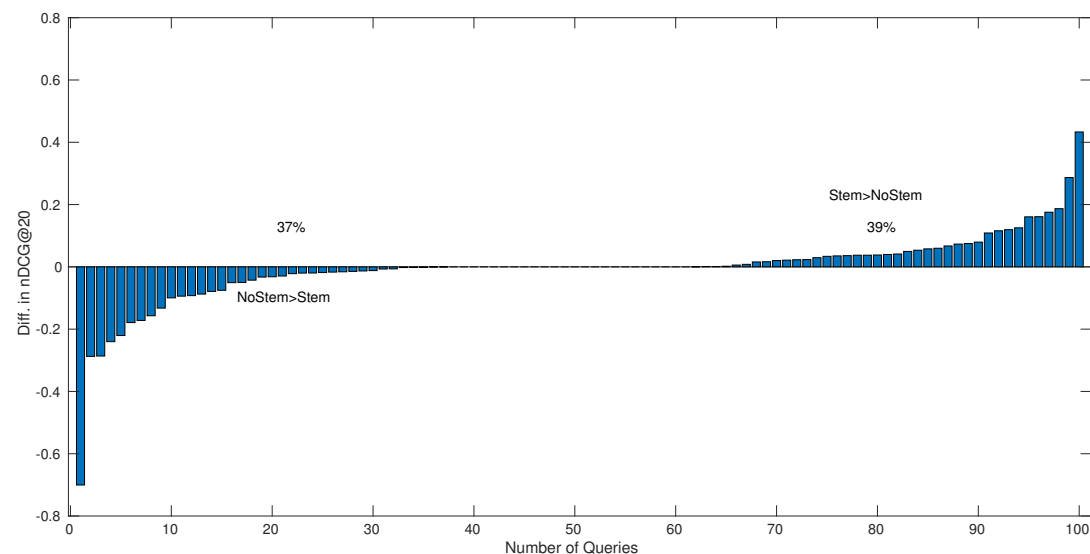



Figure 1. The NTCIR-13 WWW-1 Track for 100 queries, that the per query performance score difference between BM25 with stemming (STEM) and BM25 without stemming (NOSTEM), where stemming algorithms is Krovetz stemmer.

MOTIVATION

The factors of per query performance variability between retrieval strategies are extensively studied at the reliable information access workshop for robust retrieval (Harman and Buckley, 2004; Buckley, 2004). To identify those factors, Buckley (2004) collates the reasons why IR systems fail for individual queries into ten categories of which stemming is a category identified as general technical failures. On this account, Harman (1991) says that queries are affected by stemming positively or negatively:

> "Although individual queries were affected by stemming, the number of queries with improved performance tended to equal the number with poorer performance, thereby resulting in little overall change for the entire test collection."

For instance, an IR system in which stemming is applied has produced a score of 0.1568 for the query *poker tournaments* (TREC Web Track 2009 QueryID:17), while the system without stemming has produced a score of 0.4132 for that query. However, applying stemming to the query *mothers day songs* (TREC Web Track 2011 QueryID:132) has provided higher performance than without applying stemming in the IR system.

On the other hand, Buckley (2004, 2009) says that it may be more important to determine what strategies should be applied to which queries, rather than developing new IR strategies. In this context, stemming can be applied in a selective, binary classification manner, in such a way to improve the robustness of an IR system, so as to alleviate the performance variability across queries. Thus, a selective approach to stemming can avoid the issue mentioned in the work of Harman (1991) to a certain degree.

Fig. 1 shows, for 100 queries from the NTCIR-13 WWW-1 Track (Luo et al., 2017), that the within query difference in performance score between the run BM25 with stemming (STEM) and the run BM25 without stemming (NOSTEM), where stemming algorithms is Krovetz stemmer (Krovetz, 1993). The *x*-axis of the figure shows query number and the *y*-axis the score difference of the two runs for the corresponding query. On the left side of the figure, the queries within which NOSTEM has higher scores than STEM are shown (37%), and on right side the queries within which STEM > NOSTEM (39%). The queries on which STEM = NOSTEM are shown in the middle of the figure (24%). On average, the BM25 run with stemming, STEM has an nDCG@20 score of 0.3709 and the run NOSTEM 0.3755. Here, the difference between two average nDCG@20 scores can statistically be attributed to chance fluctuation (*p*-value = 0.70 for the paired *t*-test). Considering the observed difference in average performance scores, it would appear that, even for a recent official evaluation effort, NTCIR-13 WWW-1 Track, queries are

affected by stemming positively or negatively as mentioned in the work of Harman (1991). This suggests that Harman's argument still keeps its strength.

In this context, it can be said that, following the Buckley's argument, determining in advance which queries should be stemmed, that is, selective stemming, would be a solution. Note that here, a perfect selective approach to stemming, an oracle method could achieve an nDCG@20 score of 0.4041 on average, which is statistically better in retrieval performance than both STEM (p-value = 0.0004) and NOSTEM (p-value = 0.00002). In conclusion, the aim of this study is to minimize the risk of failure by improving the robustness of an IR system by automatically deciding whether stemming should be applied to a given query.

RELATED WORKS

Selective IR is addressed by applying a particular retrieval strategy to the given query. These strategies are used in the pre-retrieval to post-retrieval phases of the IR system, depending on the applied technique. This section primarily presents studies on selective IR applied in phases of an IR system. Then, we review the stemming algorithms in the literature by grouping them into rule-based and corpus-based. Selective approaches to stemming in IR systems is reviewed in the next section. The last section gives the differences from prior work.

Selective IR

Selective approaches can be applied to any phase of an IR system, as each phase usually encompasses different techniques that can be selected. Indexing a document collection is the first phase of the IR system in which a selective approach has the potential to be applied. Large-scale document collections are partitioned into several topically homogeneous groups named shards. Searching is only executed on a few shards that are postulated to involve relevant documents for a given query, which is called selective search. This strategy aims to reduce the retrieval cost for the query by using only a small piece of collections and preserving retrieval effectiveness as possible as that of an exhaustive search. In this respect, resource selection algorithms and techniques to select shards are proposed (Kulkarni and Callan, 2015; Kim et al., 2016, 2017). Each sub-component of IR systems affects the retrieval performance on a query basis. Considering the studies, it is seen that this research area has rich literature.

As part of an IR system, term weighting models affect per query performance variance, and queries do not benefit equally from term weighting models. Therefore, selective approaches to term weighting model selection (He and Ounis, 2003, 2004b; Arslan and Dinçer, 2019) are presented to alleviate performance degradation across term weighting models.

For another part of an IR system, the selective approach to Learning-to-Rank (LTR), which reranks retrieved documents with a learned model, is presented (Peng et al., 2010; Balasubramanian and Allan, 2010; Ghanbari and Shakery, 2019). As in the term weighting models, different queries take advantage of each ranking function differently and selective methods are studied to decide appropriate function on a per query basis. In addition, the researchers in the work (Tonellotto et al., 2013) proposed a selective pruning framework. Their work determines if the result list of the query should be pruned aggressively.

Query expansion techniques append new terms to the query to increase the recall of an IR system by matching more documents. However, it is not always the case that any query makes use of the appended terms in terms of retrieval effectiveness. Hence, selective approaches to query expansion are proposed to determine whether it should be applied (Amati et al., 2004; Cronen-Townsend et al., 2004; Hauff et al., 2010) and also to select the terms being added to the query (Cao et al., 2008a; Saleh and Pecina, 2019).

Optimization of an IR system configuration that seeks to maximize the performance of the IR system by predicting appropriate techniques is a research area in IR. The studies in this area (Bigot et al., 2015; Deveaud et al., 2018; Mothe and Ullah, 2021) aim to predict the most appropriate combination of the IR system components ranging from indexing to document ranking. The common point of these studies is to deal with all parts of the system in its entirety.

Stemming

Stemming is a remedy for vocabulary mismatch, and the application of stemming to an IR system takes place in the pre-retrieval phase. Being an important recall enhancement tool for the preprocessing phase of an IR system has made stemming a rich place in the literature. In this context, stemming methods can be broadly classified into two groups: *i*) rule-based stemming methods and *ii*) corpus-based

stemming methods. Rule-based stemmers such as Lovins stemmer (Lovins, 1968), Krovetz stemmer (Krovetz, 1993), Porter stemmer (Porter, 1997), and Paice/Husk stemmer (Paice, 1990) transform terms to their morphological roots using language-specific rules. Specifying the rules in a particular language needs expertise in that language. In addition, several linguistic resources can be used in developing a rule-based stemmer. Once the rule is created for a language, it can be used in any corpus without additional processing, making it ease of use. For different kind of languages (i.e. Arabic (Al Kharashi and Al Sughaiyer, 2002; Abuata and Al-Omari, 2015), Croatian (Ljubešić et al., 2007), Urdu (Gupta et al., 2013), Bengali (Sarkar and Bandyopadhyay, 2008; Mahmud et al., 2014), Marathi (Patil and Patil, 2017)), rule-based stemming approaches are presented in literature. Similar to the rule-based method, removing suffixes or prefixes like pluralization handling is a stemming technique. On the other hand, corpus-based stemmers construct the conflation sets, involving the morphological variants of terms, from a given corpus without requiring any linguistic knowledge. Lexicon analysis with string processing, character n-gram based analysis, co-occurrence of words, and context analysis on a given corpus are common techniques in corpus-based stemming. Lexicon analysis based stemmers group the related words in the corpus words. This strategy is usually performed by the operations such as finding suffixes, suffix stripping, string distance, etc. (Oard et al., 2001; Goldsmith, 2001; Paik et al., 2011a). In the character n-gram based method, adjacent characters in a length of n from the words in a corpus are considered to have less frequency whereas the variants have higher frequencies (McNamee and Mayfield, 2004; Ahmed and Nürnberger, 2009; Pande et al., 2018). Also, various studies on corpus-based stemming using co-occurrence analysis and machine learning techniques are presented (Paik et al., 2011b, 2013; Brychcín and Konopík, 2015). These methods analyze the co-occurrence or context of the basis form of the words in a corpus. In this regards, lexical and co-occurrence similarities are usually applied to discover morphologically related words. For instance, the work of Singh and Gupta (2019) also employed suffix pair frequency and graph-based clustering besides lexical and co-occurrence similarity in order to construct the conflation sets. In another work, researchers applied Hidden Markov Model to produce stems of the words(Bölücü and Can, 2019).

Recently proposed method (Singh and Bhowmick, 2022) uses neural network to predict co-occurrence similarity between a query term and its potential morphological variant. Potential or candidate variants are initially determined by using lexical similarity of the words in the corpus.

Selective stemming

From the perspective of selective stemming, several studies are presented to decide whether stemming should be applied to an IR system on a per query basis. In the works of Harman (1987, 1991), the selection process was carried out on the basis of two criteria, query length, and term importance. However, the results of the experiments provided no evidence indicating significant improvement in retrieval effectiveness on average. In another work of Harman (1991), the selection was simply based on a threshold: that is, if the query length is shorter than ten terms, stemming is applied; otherwise, no-stemming is applied. Machine learning based selection methods was also proposed in the work of Chin et al. (2010). The method in the work employs Support Vector Regression models built on the query features to select an appropriate text normalization technique among stemming, depluralization, and without any text normalization. On the other hand, advanced optimization algorithms are also used for selective stemming. In this context, a selective approach to stemming based on a genetic algorithm leveraging query performance predictors is used as well (Wood, 2013). In addition to selecting a stemming method, there are different works on selective stemming. Stemming may be applied to an IR system at run time by expanding the query with the morphological variants of its query terms. Thereby, the method that select the morphological variants from the set of candidate variants for a given query is a selective approach to stemming (Croft and Xu, 1995; Peng et al., 2007; Cao et al., 2008b). Particularly, in the work of Tudhope[1], query expansion was accomplished by means of a filtering strategy that filters out those morphological variants of the query terms that have not the same semantic meaning. This query expansion technique was examined using TREC-4 Ad Hoc task queries, and the results of the experiments show that it improves the accuracy of the stemming algorithm in use. Similarly, in the work of Cao et al. (2008b), a query expansion technique that expands the query on the basis of the expected effect of individual variants on retrieval effectiveness was introduced. The study also adopted a further technique that uses the language model to decide the morphological variants of query terms that best fit the query context. The results of the experiments show

---

[1]https://cs.uwaterloo.ca/research/tr/1996/31/cs-96-31.pdf

a significant improvement over the not stemmed system in retrieval effectiveness on GOV2 Web Track and several TREC Ad Hoc track queries. However, the methods in the experiments couldn't show the same performance as the traditional stemming method that expands the query with all morphological variants of the query terms. The similarity between the query term and its morphological variants may be determined by word embeddings. Word embedding is a vectorial representation of a term so as to measure contextual similarity between given terms. String similarity and contextual similarity are employed to select morphological variants (Basu et al., 2017). The researchers in the work (Roy et al., 2017) used local and global word embeddings to measure contextual similarity within term variants and filter them to create final clusters. Furthermore, in the same line of research, the selection of the term variants based on the language models was also investigated in the work of Peng et al. (2007) utilizing an occurrence analysis of query term variants on the corresponding document collection.

Differences from prior work

The selective stemming method presented in this paper considerably extends previous works (Harman, 1987, 1991; Chin et al., 2010; Wood, 2013) in which we follow the same research line. One of the limitations of these works is the size of the query sets: the number of tested queries is relatively smaller than the query sets that have been brought to the literature in total recently. Another limitation is the decision methods: Even if some of the works build a machine learning model, the features used in the model must be able to discriminate the queries as possible. Features in the works usually could not cover the term frequency distribution yielded by an IR system where stemming is applied. However, stemming does not behave collection and document frequencies of the terms in a corpus similarly since the frequencies of some terms would substantially increase from the others.

Our proposed method is a binary classifier that leverages the not only query performance predictor features but also the derived features from the fluctuation in an increase of the term frequencies for the systems with and without stemming. To our best knowledge, the proposed work in this paper is a pioneering work that quantifies the inter-relation of within query term specificity (Spärck Jones, 1972; Robertson, 2004; Church and Gale, 1999) and derives query features from that information. Those features usually leverage the differences in term specificity for both with and without stemming. This method is evaluated on a diverse set of standard IR test collections using out-of-the-box retrieval configuration. The test collections used in the experiments involve eight query sets related to four comprehensive document collections that broadly include Web documents and also newspaper journals. In addition, we perform a risk-sensitive evaluation to determine the robustness of the proposed selective stemming method. This evaluation indicates that the average performance enhancement by the method is spread to the queries.

PROPOSED SELECTIVE APPROACH TO STEMMING

The pioneering works on selective approach to stemming are the works of Harman (1987, 1991). Those works employ only one selection criterion to make the binary decision (i.e. stemming vs. no-stemming). That selection criterion is the length of queries measured in terms of non-common terms, i.e., the number of non-common query terms. Stemming is basically applied to those queries of which the length is less than ten terms. Similarly, in the work of Wood (2013), a selective approach to stemming is also introduced on the basis of a binary decision process. The decision relies on a genetic algorithm utilizing query performance predictors, including inverse document frequency, inverse collection term frequency, average collection query scope, etc. However, the results of the experiments presented in the mentioned works show that the introduced selective approaches provide little or no improvement on the average retrieval effectiveness of the system with stemming. The proposed work in this paper is also a binary classifier for a selective approach to stemming, and it is based on a supervised classification technique known as the k-Nearest Neighbor Classifier. The classifier employs a set of query performance predictors introduced in the IR literature and a new set of features derived from differences in the term frequencies of the systems participating in the selection.

The frequency distribution of a query term on a given document collection would generally change depending on whether or not stemming is applied to that term. In this study, we assume that the difference in term frequency distribution obtained after applying stemming to a query term can be used as a criterion to decide whether stemming should be applied or not. Therefore, in our study, the features presented in the literature and produced based on our assumptions are used. The query performance predictors that are borrowed from the literature are of pre-retrieval type (Carmel and Yom-Tov, 2010), including minimum

286 IDF ratio over maximum IDF ratio (Gamma), query scope (Omega) (He and Ounis, 2004a), maximum
IDF and AvgSCQ (Zhao et al., 2008). In addition to those four features, we introduce the following six
features derived from the frequency distributions of query terms, given as follows:

- **AvgIncDF**: Stemming increases the document frequency (DF) of a particular term by unifying
  the posting lists of its morphological variants in IR systems. The increase may not be the same
  degree for all terms in the query. This feature basically measures the average increase in document
  frequency for a given query, as in Eq. 1.

$$\frac{1}{|q|}\sum_{i=1}^{|q|}\frac{DF_{i_{Stem}} - DF_{i_{NoStem}}}{DF_{i_{NoStem}}} \tag{1}$$

In the equation $|q|$ is the number of query terms in the given query.

- **MaxWeightedIncDF**: Inverse document frequency measures the term specificity of a term in a
  document collection (Spärck Jones, 1972; Robertson, 2004). It can be supposed that an importance
  value calculated by multiplying the increase in the document frequency and term specificity would
  be an indicator to measure the retrieval effectiveness of applying stemming to the query terms. The
  maximum importance value among the calculated values for each query term is used as a feature,
  as given in Eq. 2.

$$Max\left\{Idf_{1_{NoStem}} \times \frac{DF_{1_{Stem}} - DF_{1_{NoStem}}}{DF_{1_{NoStem}}}, ..., Idf_{n_{NoStem}} \times \frac{DF_{n_{Stem}} - DF_{n_{NoStem}}}{DF_{n_{NoStem}}}\right\} \tag{2}$$

In the equation, n is the number of terms in the query.

- **CorrIctfRank**: After stemming is applied, in a query, term specificity such as inverse document
  frequency (Idf) or inverse collection term frequency ($Ictf$) for a given query term may relatively
  change according to the other terms in that query. We apply this observation as a feature (it takes 0
  or 1) measuring the correlation between the ranks of the term positions of the query terms. Two
  lists including term positions are constructed for both stemming and no-stemming and they are
  sorted by term specificity (i.e. $Ictf$). Here, we use the Spearman rank correlation and we assume
  that they are correlated if the correlation coefficient value is greater than 0.7.

- **Mst-Lst-Change**: After applying stemming, the least specific or the most specific term in a given
  query may change. Those changes are used as a feature. If a change occurs on the least specific or
  the most specific term, it takes 1; otherwise, 0.

- **Chi2-DF-TF**: When stemming is applied to the given query, document frequency and collection
  term frequency of the query terms may be affected more than other terms. We use $p - value$ of
  the Chi-square goodness-of-fit test to see this effect and employ it as a feature. First, we construct
  two lists involving document and collection term frequencies of the given query terms for both
  stemming and no-stemming, preserving the term positions. Since Chi-square goodness-of-fit test
  requires discrete values, a binning strategy, the Freedman-Diaconis rule, is employed to those
  frequencies to obtain a finite number of bins and groups of the terms. Thus, $p - value$ of the
  Chi-square goodness-of-fit test is obtained with using two lists and employed as a feature.

- **ModifiedSCS**: This feature is modified version of the *simplified query clarity score* proposed in the
  literature (Cronen-Townsend et al., 2002; He and Ounis, 2004a).

$$\sum_{Q} P_{ml}(w|Q) \cdot \ln(\frac{P_{ml}(w|Q)}{P_{coll}(stem(w))}) \tag{3}$$

In Eq. 3, $P_{ml}(w|Q)$ is the maximum likelihood of the query model as in the proposed definition in
the literature. It is given by qtf/ql, where qtf is the frequency of a query term in the query and
ql is the query length. $P_{coll}(stem(w))$ is the collection model. Whereas it is defined by $P_{coll}(w)$,
which is given by $tf_{coll}/token_{coll}$, in the literature, it is modified by $tf_{coll}/tf_{coll}(stem(w))$, where

tf_{coll}, $token_{coll}$, and $tf_{coll}(stem(w))$ are the collection term frequency, the number of total terms in the collection, and the collection term frequency of a stemmed query term respectively. The value of $tf_{coll}(stem(w))$ can be calculated by summing collection term frequencies of the terms in the conflation set of the query term at running time.

The labels used in the classifier are derived from retrieval performance scores of both no-stemming (NoStem) and used stemming algorithm in selection. The labeling process using the retrieval performance scores of the queries proceeds as follows: If the retrieval performance score of the query in stemming is higher than the score in no-stemming, the label of the query is assigned as "1" for stemming; otherwise, "0" for NoStem. Furthermore, the queries with the same retrieval performance score for both no-stemming and stemming, tie queries, are discarded in the training phase. The remaining queries are used to build the training model. Consequently, the proposed classifier decides whether or not stemming should be applied for a given query.

EXPERIMENTAL SETUP

This section details the setup of the search engine tool and document collections in use. We have used an open-source search engine tool to perform retrieval experiments (Sect. Experimental system). In addition, the experiments are carried out using large-scale document collections and their corresponding query sets (Sect. Benchmark collections). The proposed selective approach is evaluated with normalized Discounted Cumulative Gain (nDCG) (Järvelin and Kekäläinen, 2000, 2002) at top 20 documents in the result list. Since the user naturally expects highly relevant documents to be at the top of the result list, especially for the Web collections, the evaluation metric meets this expectation by assigning more scores to the top-ranked relevant documents during the evaluation process. Sect. Baselines describes state-of-the-art baseline methods compared and evaluated in this work.

Experimental system

The IR community addresses the reproducibility of IR experiments and encourages the authors to conduct their experiments following the adopted standards as much as possible (Arguello et al., 2016; Lin et al., 2016; Voorhees et al., 2016). In this respect, details of the experimental system are given for reproducibility of the performed experiments. Source codes of the experiments are made publicly available on GitHub repositories[23].

We use Apache Lucene (Białecki et al., 2012), an open-source search engine platform that is developed for commercial purposes, to perform IR experiments in this study. Although widespread use in the industry is also gradually accelerating its use in academic research (Azzopardi et al., 2017). Indexing HyperText Markup Language (HTML) documents with Lucene, the documents are stripped from their HTML tags using jsoup[4] library so as to get plain text blocks of a given HTML document. The final text block to be indexed is obtained by combining the title and body text blocks extracted by the jsoup library into one unstructured text block. The combined text block is processed with `StandardTokinezer` and `LowerCaseFilter` of Apache Lucene without employing stemming and stopword removal in the indexing process.

Instead of keeping indexes separately for each stemming algorithm in this study, the query time stemming approach (Peng et al., 2007; Cao et al., 2008b) is utilized by means of applying `SynonymGraphFilterFactory` of Apache Lucene to the corresponding documents and the given queries. To perform this functionality, query time stemming, the search tool requires a set of morphological variants of the words to be stemmed. For this purpose, morphological variants of each query term, generated by the concerned stemming algorithm, are provided to the system. As a result, the IR system retrieves the documents for a given query according to a particular stemming algorithm over a single index in Apache Lucene. In addition to the stemming procedure mentioned, BM25 term weighting model in Terrier software[5] is adapted to Lucene[6] version 7.7.0 platform.

Table 1. Description and number of queries for Tracks according to document collections in the experiments.

Collection	Track	Label in Experiment	Number of Queries
WSJ	Ad Hoc 1,2 & 3	WSJ	150
GOV2	Terabyte 2004, 2005 & 2006	GOV2	149
	Million Query 2007	MQ07	1,524
	Million Query 2008	MQ08	564
CW09B	Million Query 2009	MQ09	562
	Web 2009, 2010, 2011 & 2012	CW09B	197
CW12B	Web 2013 & 2014 Tasks 2015 & 2016	CW12B	185
	We Want Web 13 & 14	NTCIR	180

Benchmark collections

The ClueWeb09 collection includes about 1 billion Web documents collected between January and February 2009. Category B subset of the collection includes about 50 million English Web pages. This subset has been used in TREC Web Tracks ran through 2009 to 2012 (CW09B) and Million Query Track 2009 (MQ09). The ClueWeb12 collection includes about 733 million English Web documents collected between February 10, 2012 and May 10, 2012. Uniformly extracted 7% sample of the collection is named Category B13. This collection, ClueWeb12-B13, has been used in TREC Web and Tasks Tracks 2013-2016 (CW12B) and NTCIR (Luo et al., 2017; Mao et al., 2019) We Want Web Tracks 13&14&15 (NTCIR). The GOV2 (Clarke et al., 2004) collection involves about 25 million Web documents from the `.gov` domain. TREC Terabyte Tracks 2004-2006 (GOV2), Million Query Tracks 2007 (MQ07), and 2008 (MQ08) have been conducted using GOV2. Wall Street Journal collection contains about 173 thousand newspaper articles in the TIPSTER collection of disk 1 & 2. This collection has been used in TREC 1-2-3 Ad Hoc Tasks. The experimental evaluations are carried out on a wide range of standard TREC datasets and their corresponding tracks. The number of queries for each track is summarized in Table 1.

Baselines

The proposed method decides whether stemming should be applied or not on a query basis. Thereby, two systems participating in this selection are our baselines. *NoStem* refers to the system in which any stemming method is not applied, and it is the mandatory system involved in the selection. Another one is the system in which one of the *KStem* (Krovetz, 1993), *Porter* (Porter, 1980), and *Lovins* (Lovins, 1968) stemming algorithms is applied. Hereby, we have performed the experiments by the combinations of NoStem and stemming algorithms: *NoStem-KStem*, *NoStem-Porter*, and *NoStem-Lovins*.

EVALUATION OF THE SELECTIVE APPROACH TO STEMMING

The proposed selective approach to stemming is evaluated by employing common IR practice. To accomplish the practice, BM25 term weighting model (Robertson and Zaragoza, 2009), which is an out-of-the-box option in many IR works, is used in the evaluation of the retrieval effectiveness. Furthermore, the stop-word removal process is not utilized during indexing the documents. The proposed selective approach uses k-Nearest Neighbors classification algorithm to decide whether stemming should be applied, where k is chosen as 11 and Minkowski distance with exponent value set to 3 is used to find the closest neighbors. Leave-one-out cross-validation method is employed to evaluate the classifier so that each query is used as a test query during the evaluation process (Arlot and Celisse, 2010).

---

[2]https://bitbucket.org/gokhanc/lucene-clueweb-retrieval/src/master/
[3]https://github.com/gokhanc90/matlabIRexperiments
[4]https://jsoup.org/
[5]http://terrier.org/
[6]https://lucene.apache.org/

Table 2. The table presents the nDCG@20 scores of the systems for the query set collections.

Collection	# Queries	NoStem	KStem	SelKStem	NoStem	Lovins	SelLovins	NoStem	Porter	SelPorter
CW09B	197	0.1606	0.1523	**0.1640**^{N,B}	0.1606	0.1460	**0.1634**^{B}	0.1606	0.1497	**0.1563**
CW12B	185	0.0781	0.0858	**0.0872**^{N}	0.0781	0.0807	**0.0849**^{N}	0.0781	0.0872	**0.0874**^{N}
NTCIR	178	0.2946	0.2956	**0.3051**^{N,B}	0.2946	0.2681	**0.2934**^{B}	0.2946	0.2839	**0.2972**^{B}
GOV2	149	0.3486	0.3697	**0.3707**^{N}	0.3486	**0.3670**	0.3611	0.3486	0.3839	**0.3885**^{N}
WSJ	148	0.3313	**0.3661**	0.3576^{N}	0.3313	**0.3485**	0.3417^{N}	0.3313	**0.3684**	0.3657^{N}
MQ07	1520	0.2253	0.2313	**0.2321**^{N}	0.2253	0.2227	**0.2264**	0.2253	**0.2366**	0.2315^{N}
MQ08	562	0.2623	0.2607	**0.2631**	0.2623	0.2431	**0.2586**^{B}	0.2623	**0.2631**	0.2609
MQ09	562	0.2508	0.2543	**0.2546**	0.2508	0.2433	**0.2479**	0.2508	**0.2552**	0.2549

Table 3. Average character length of the distinct terms in the queries.

Tracks	Sum of Distinct Term Length	# Distinct Terms in Queries	Average
Web 2009, 2010, 2011 & 2012	2,457	419	5.86
Web 2013 & 2014, Tasks 2015 & 2016	2,705	456	5.93
Terabyte 2004, 2005 & 2006	2,791	413	6.76
Million Query 2007	18,616	2,824	6.59
Million Query 2008	9,408	1,475	6.38
Million Query 2009	7,113	1,140	6.24
We Want Web 13 & 14	2,458	396	6.21
Ad Hoc 1,2 & 3	3,159	447	7.07

Retrieval results

Table 2 lists the average nDCG@20 scores of the systems for the query set collections. The proposed method is named by appending the prefix *Sel* keyword to its baseline stemming algorithm. The highest scores are indicated in boldface. The highlighted scores are the values that NoStem has the best score against the scores of selective method and the corresponding stemmer. The collections of query sets and the number of queries in those sets are given in the first and second columns respectively. We have discarded the queries where at least one of its terms is not found in the document collection. For example, the term *tetacycline* (TREC 2008 Million Query Track QueryID:16625) does not occur in the GOV2 dataset. The remaining columns show the performance scores of the systems and selective method. *B* refers that the performance difference between the proposed method and the baseline stemmer is statistically significant with a *p*-value of < 0.1. Similarly, *N* denotes that the performance difference between the proposed method and NoStem is statistically significant with the same *p*-value.

NoStem has the highest scores for NTCIR, MQ08, and MQ09 collections in Lovins, and for CW09B collection in Porter. The proposed method with KStem produces the highest scores on average for each corresponding collection of query sets except WSJ. Furthermore, it produces a statistically significant improvement in average performance over NoStem for the collections CW09B, CW12B, NTCIR, GOV2, WSJ, MQ07, and significant over KStem for the CW09B and NTCIR collections. However, the proposed method yields higher performance scores than its baseline stemmer (Lovins) except for GOV2 and WSJ collections. The improvements are statistically significantly higher than its baseline stemmer for CW09B, NTCIR, and MQ08 while it is significantly higher than NoStem for CW12B and WSJ. The proposed method with Porter yields the highest scores for CW12B, NTCIR, and GOV2. For the collections CW12B, GOV2, WSJ, and MQ07, the performance of the selective method is statistically significantly higher than NoStem and significantly higher than baseline stemmer for NTCIR.

The most important conclusion from the Table 2 is that the selective approach systematically improves the average retrieval performance of the single system for most of the query sets. Another important result is that the approach does not produce a significantly worse performance score than any single system, even if various collections of query sets are used. On the other hand, selective method could not succeed for WSJ collection. All stemming algorithms produce the highest score for this collection. The reason for this situation can be explained by interpreting Tables 3 and 4. Table 3 lists the average length of the distinct terms in the queries for each query set. It can be easily seen that average length of the terms are close to

Table 4. Average character length of the distinct terms in the corpus of the document collections

Corpus of Document Collection	Sum of Distinct Term Length	# Distinct Terms in Corpus	Average
WSJ	1,651,354	212,146	7.78
GOV2	95,309,991	10,440,851	9.13
CW09B	443,897,471	44,114,780	10.06
CW12B	556,007,008	52,065,545	10.68

Table 5. The classifier accuracy of the selective approach is presented for each stemmer separately.

Collection	True Predicted		Actual		Tie	Accuracy (%)
	NoStem	KStem	NoStem	KStem		
CW09B	50	17	64	37	96	66
CW12B	20	33	38	45	102	64
NTCIR	28	49	60	67	51	61
GOV2	19	56	52	75	22	59
WSJ	20	47	49	72	27	55
MQ07	223	165	366	343	811	55
MQ08	91	61	154	132	276	53
MQ09	89	67	149	136	277	55

Collection	True Predicted		Actual		Tie	Accuracy (%)
	NoStem	Lovins	NoStem	Lovins		
CW09B	72	12	83	45	69	66
CW12B	21	34	53	52	80	52
NTCIR	62	19	87	60	31	55
GOV2	15	50	51	74	24	52
WSJ	4	43	42	60	46	46
MQ07	250	182	444	385	691	52
MQ08	156	38	201	136	225	58
MQ09	113	55	189	145	228	50

Collection	True Predicted		Actual		Tie	Accuracy (%)
	NoStem	Porter	NoStem	Porter		
CW09B	59	13	75	48	74	59
CW12B	22	38	47	57	81	58
NTCIR	40	44	75	71	32	58
GOV2	16	71	51	86	12	64
WSJ	19	59	50	80	18	60
MQ07	185	244	392	430	698	52
MQ08	96	71	176	162	224	49
MQ09	97	78	181	155	226	52

420   each other. Table 4 lists the average length of the distinct terms in the corpus of the document collections.
421   WSJ document collection and its corresponding query set Ad Hoc Tracks have close term length on
422   average but other document collections have higher average term length than their corresponding query
423   sets. One of the differences between WSJ and other document collections is that WSJ is a newspaper
424   collection while others include Web documents. Other difference is the number of distinct terms in corpus.
425   WSJ includes quite a few distinct terms compared to other corpus. Therefore, considering the scarcity of
426   documents and the number of distinct terms in the WSJ corpus, it benefits from stemming algorithms
427   alleviating the mismatching problem since more documents can be scored, and more relevant documents

can be accessed. Since other corpora contain a large number of documents, it can be considered that this problem will arise less frequently. When stemming is applied, it is usually expected that irrelevant documents will be included in the result list in a collection with so many documents, and this will hurt performance in some queries. This argument can be made as an inference from the highlighted cases in Table 2. For instance, NoStem has the highest scores for the query set CW09B against Porter and for the query sets NTCIR, MQ08, and MQ09 against Lovins.

Fig. 2 shows the multiple comparisons after Friedman's test for NoStem, selective method, and baseline stemmers. It tests column effects(i.e., stemming algorithms, selective method, and NoStem) after adjusting for possible row effects (i.e., queries). The test is appropriate for the multiple comparisons of results of IR experiments since the compared methods are under study and queries do not have any interaction with each other. Tukey's honestly significant difference criterion (Tukey's HSD) is employed for multiple comparisons. The figure only shows CW09B and CW12B collections and the remaining collections are given in appendix Sect. Multiple comparisons. For CW09B, it appears that the selective method is statistically different from all its base stemmers, but we could not observe the case for other query sets. The selective method with Porter in several cases, such as GOV2 and WSJ, is significantly different from NoStem. Another important point is that the selective method is significantly no worse than the baselines in the experiments, which consisted of the combinations of eight query sets and three stemming algorithms.

The classifier accuracy of each collection is listed in Table 5. The table contains the correctly predicted and actual number of queries for NoStem and each stemming algorithm. The number of queries with the same performance scores for both NoStem and corresponding stemming is in the Tie column. It is inquired in *RQ1* to what degree the proposed selective approach is accurate in predicting the queries that stemming should (not) be applied. The proposed classifier for selective stemming produces moderate accuracy scores for predicting whether or not stemming should be applied. However, the accuracy results only show the classifier performance by ignoring the differences between retrieval scores of the selections. Therefore, the average retrieval system performance score and classifier accuracy needs to be considered together. Thus, Table 2, plots in risk-sensitive analysis (Fig. 3), and plots in per query performance analysis (Fig. 4, Fig. 5) show the positive effects of this classifier on the average performance of the retrieval systems by means of the performance evaluation metric for CW09B and CW12B query sets. In addition, the importance of those features in selective stemming is discussed in Sect. Feature analysis by presenting in Fig. 6.

We have attached the figures of the remaining query sets to the Sect. Appendix to keep the reading flow. The plots for the statistical test are given in Fig. 7. The related plots to the risk-sensitive and per query performance analysis are presented in Fig. 8 and Fig. 9, respectively. Finally, feature importance plots are presented in Fig. 10.

Risk-sensitive analysis

In a risk-sensitive evaluation for the robustness of an IR system, the term risk refers to the risk of retrieval effectiveness of an IR system for a given particular query worse than the effectiveness of a baseline system for that query; otherwise, it refers to reward. A control parameter $\alpha > 0$ penalizes the system with respect to the risk-reward trade-off giving more weight to the risk, and where $\alpha = 0$ refers to the risk and reward having equal weight. To measure the robustness of the selective approach, we utilize *TRisk* (Dinçer et al., 2014) risk-sensitive evaluation measure. *TRisk* is a measure based on hypothesis testing with the identification of queries that commit a significance level of risk. This risk measurement makes use of the linear transformation of t statistic used in the Student's t-test. The threshold values for *TRisk* are -2 and $+2$. An IR system with *TRisk* < -2 at $\alpha = 0$ is under a risk. For an IR system with *TRisk* > 0, the system is counted in favor of reward. The performance of the system is considered statistically significantly better than the baseline if the values *TRisk* > 2.

RQ2 inquires the contribution of the proposed selective approach with respect to robustness in retrieval effectiveness of the IR systems that employ stemming. In this respect, Fig. 3 shows the robustness of the proposed selective approach against single systems in *TRisk* risk-sensitive evaluation metric. In the figure, we have presented CW09B and CW12B results together and the remaining query sets are appended to the Appendix . The plots in the figures compare the robustness of selective method and its baseline stemmers against NoStem. The proposed method is more robust than the system in which the baseline stemming algorithm is applied. This improvement in robustness has been achieved for almost all experiments.

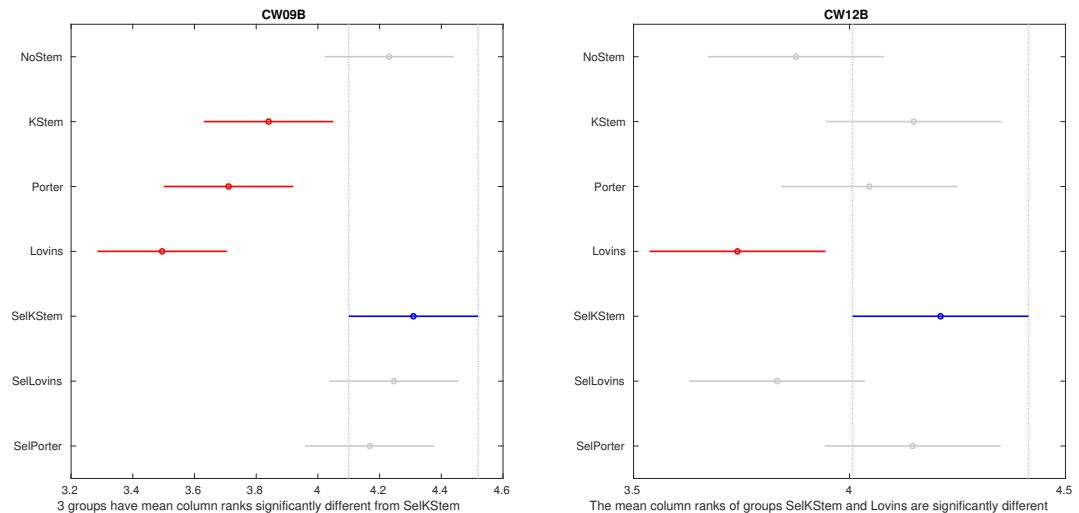



Figure 2. Multiple comparisons based on performance scores of the stemmers in IR systems for CW09B (a) and CW12B (b)

(a) TRisk comparisons of proposed method and baseline stemmers against NoStem for CW09B

(b) TRisk comparisons of proposed method and baseline stemmers against NoStem for CW12B

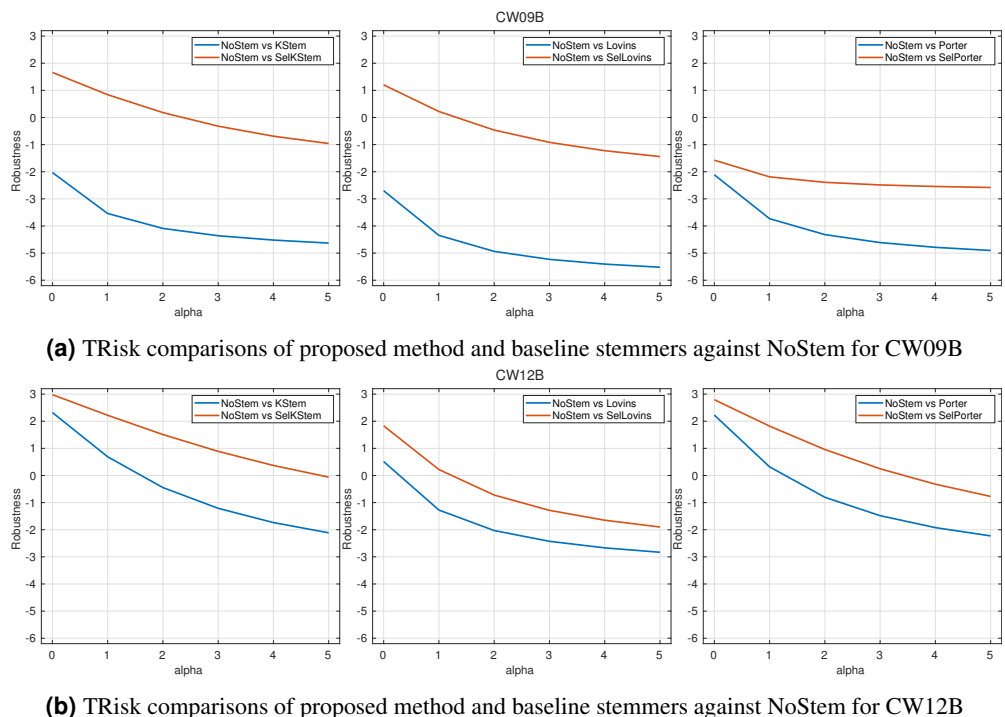



Figure 3. TRisk comparisons of proposed method and baseline stemmers against NoStem for α parameter from 0 to 5.

For the increasing values of α, we observe that selective method maintains the robustness for the query sets. We observe that the minimum *TRisk* value among the query sets is -10.03 for the KStem baseline, but it is -4.98 for selective method. The similar results are valid for Lovins and Porter baselines. The selective method has a minimum *TRisk* value of -7.82, while the baseline Lovins has a minimum value of -12.61. Furthermore, similar results can be obtained for the baseline Porter: the selective method and the baseline have minimum values of -6.78 and -10.30 respectively. Consequently, the proposed method has minimized the risk of failure in IR system, and those results indicate the contribution of this study.

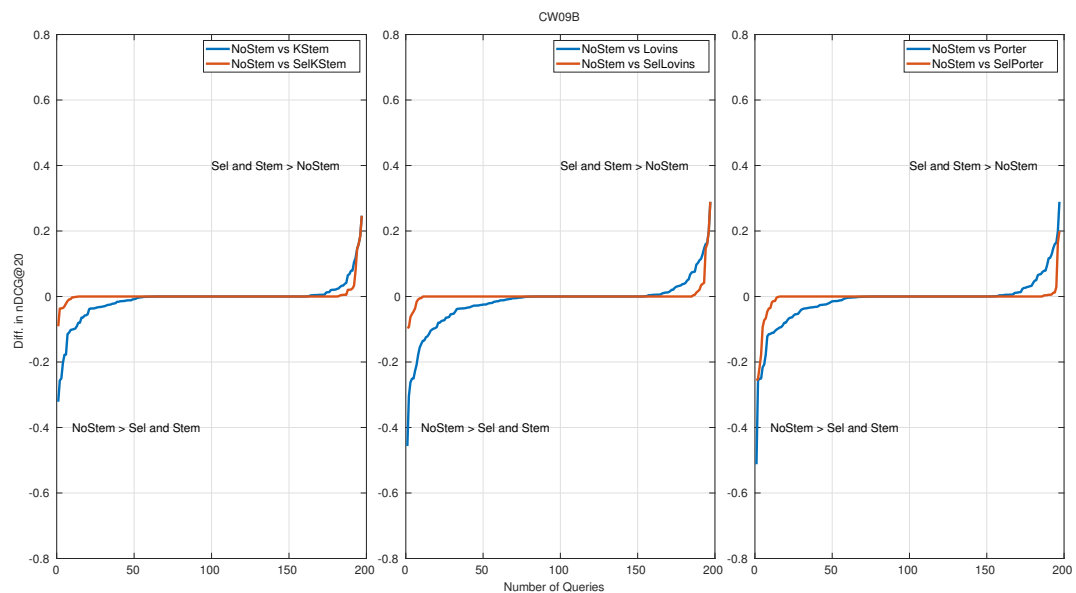



Figure 4. Per query performance differences for CW09B

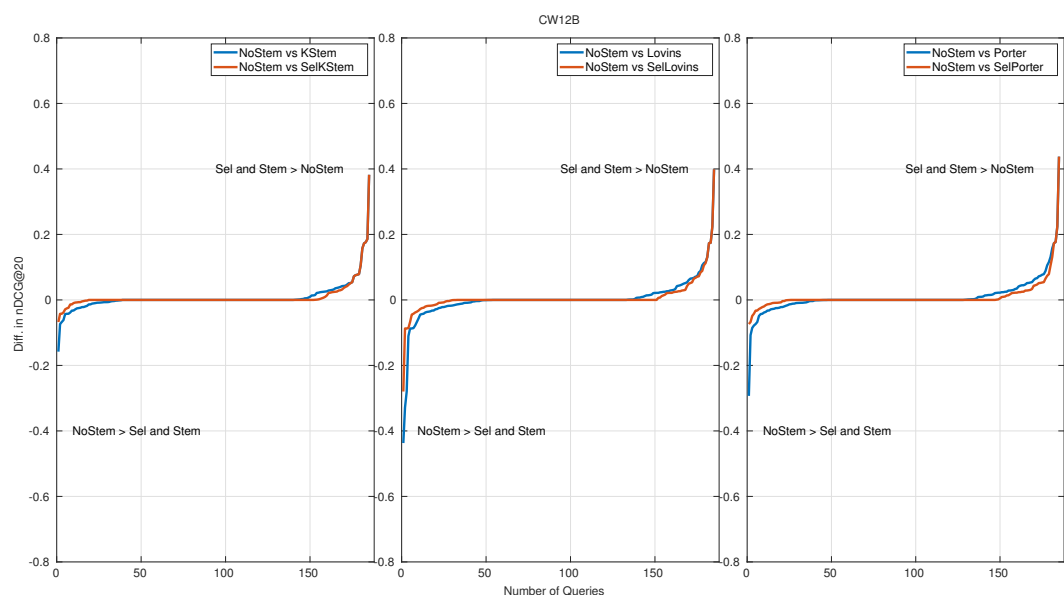



Figure 5. Per query performance differences for CW12B

Per query performance analysis

Per query performance analysis is performed to reveal the number of queries that benefit from or are hurt by the proposed method. Figures 4 and 5 show the nDCG@20 score differences on query basis for CW09B and CW12B query sets, sorted in ascending order. The remaining figures of query sets are appended to the Appendix . Each subfigure has three plots: each of them is produced for NoStem vs. Selective(Sel), and NoStem vs. each base stemmer is placed together in corresponding plot. For instance, in the first plot in Fig. 4, performance differences between NoStem and KStem, and between NoStem and SelKStem are plotted using line graph. Similarly, the second and third plots are for Lovins and Porter, respectively. The axis-y in the plot represents the performance difference for each queries. The high values in axis-y indicate a high-performance difference between systems in terms of retrieval effectiveness.

The score of the retrieval effectiveness of each query in the proposed method is one of the scores

(a) Feature Importance for CW09B

(b) Feature Importance for CW12B

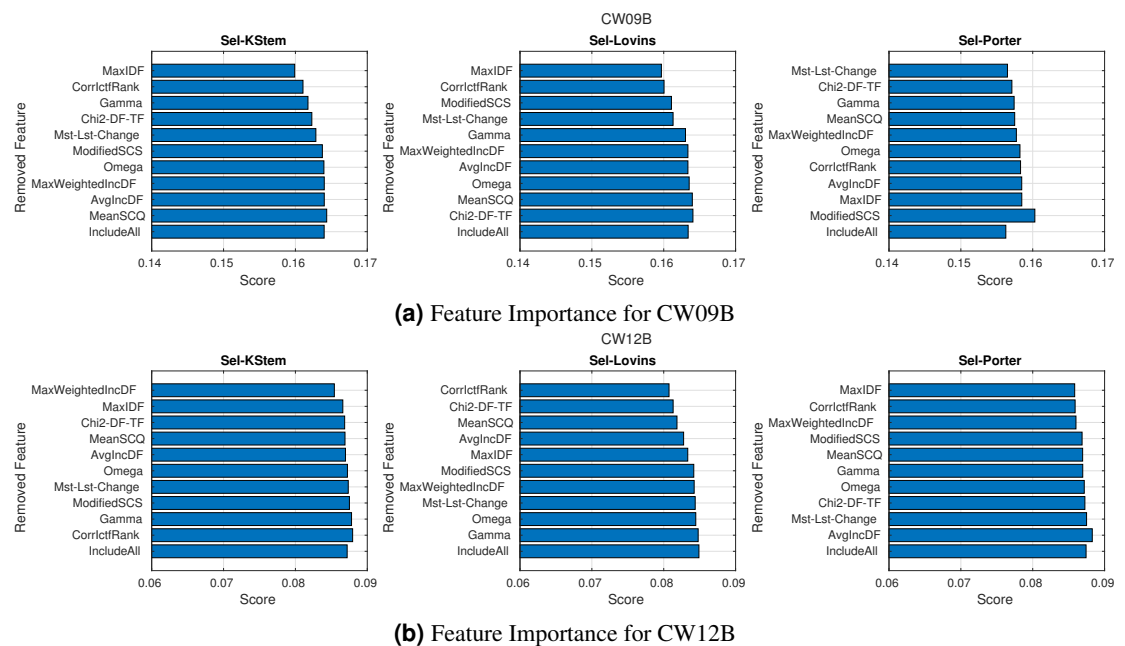



Figure 6. Feature importance according to the ablation study

produced by the participating systems. This is the reason for the increase in the tie queries (performance differences are equal to 0), which is the middle part of the graphics. According to the plots, it is seen that the selective method decreases the number of queries that are negatively affected by the stemming. In Fig. 4, a 25 percent decrease in the amount of negatively affected queries by stemming for KStem is shown when selective stemming is applied. This is also observed in 36 percent and 30 percent for Lovins and Porter respectively. The percentages for Fig. 5 are an 11 percent decrease for KStem and Porter and a 12 percent decrease for Lovins. In addition, the reason for the average performance improvement with the proposed method is not due to successful predictions in a few queries but to successful predictions across queries in general. For instance, a 0.0117 improvement in performance effectiveness is produced by selective stemming against KStem; at the same time, the percentage of negatively affected queries reduces to 25 percent for CW09B. Moreover, the method alleviates a deterioration in performance by accurately predicting as possible the queries placed in the most left part of the plots. The reduction of the areas in the lower left part of the plots reveals minimizing the risk of failure, and it also confirms the implications of the risk-sensitive analysis.

The results of proposed selective method are slightly worse for query sets of Million Query Tracks as listed in Table 2 than the IR system that Porter is applied. Part of the reason of this situation is to not accurately make prediction for a few queries where performance differences between stemming and no-stemming are the largest. For example, the Porter plots in Fig. 9f shows that selective method could not accurately predict those a few queries. This error in prediction is most likely due to machine learning method since the machine learning methods only learn from given sample of data. Thereby, the average performance of the system is affected by this situation. However, this does not harm the robustness of the proposed method, but rather maintains the robustness by minimizing the risk of failure across queries.

Feature analysis

Feature analysis investigates to what degree each feature contributes to performance. Our proposed work takes advantage of several features used in binary classifiers for selective stemming. The features have been used in 24 experiments (a combination of eight query sets and three stemming algorithms), so in this section, we have examined the impact of the features on IR effectiveness. Here, the investigation aims to discover the most and the least important features and rank them in terms of the contribution to the performance improvement of the proposed work.

In conducting feature analysis experiments, we have employed an ablation study. This evaluation ranks the features according to their importance. The features are sequentially removed from the feature

Table 6. The average and median of the ranks in the feature importance experiments are presented.

Feature	Average	Median
MaxIDF	4.6	4
MeanSCQ	4.6	4
CorrIctfRank	5.2	5
Chi2-DF-TF	5.5	6
Mst-Lst-Change	5.5	6
ModifiedSCS	5.6	6
Omega	5.7	6
MaxWeightedIncDF	5.8	6
Gamma	6	6
AvgIncDF	6.5	6

set. For each iteration, the classifier model is trained, and the average performance of the IR system is obtained. Finally, the performances of the IR system obtained by each reduced feature set are the importance of the corresponding removed feature. When the features are sorted in ascending order according to their importance, the first feature is the most important feature. Because the classifier model trained by the feature set without this feature produces the lowest performance score. Fig. 6 shows the plots for CW09B and CW12B query sets and each stemming algorithm. The plots for other query sets are given in Appendix . Feature labels are on the y-axis, and their importance scores are given on the x-axis in the plots. Also, the features are presented in ascending order according to their importance, but the last row in each plot (label IncludeAll) stands for the performance score obtained by the complete feature set. Each feature provides a different degree of contribution to the experiments. For instance, while feature MaxIDF is the most important feature for CW09B with KStem, it is one of the least important features for Porter, and also removing it improves the performance of the IR system more than the IR system using the complete feature set. Table 6 summarizes the plots according to the importance of the features. The table includes the average and median ranking of each feature in experiments. We can make several inferences from the table. In terms of most important and least important features, MaxIDF and AvgIncDF are the candidate features respectively. The average ranks of the features are around the fifth rank out of ten features. If we had a feature around the second rank on average, we could conclude that it is the core feature. Similarly, if we had a feature around the ninth rank on average, we could conclude that it can be removed from the feature set or it is the least important feature. However, according to our conducting experiments, the most important implication is that the features are of almost equal importance since they are around the fifth rank. The table supports that the features provide the robustness and performance improvements of the selective IR systems. In addition, removing some features from the feature set improves the retrieval scores as seen in Fig. 6. Those features are different for each experiments. For instance, removing MaxIDF from the feature set for the experiment conducted on the KStem baseline decreases the performance, but it improves for Porter baseline on CW09B. This is a typical situation because each stemming algorithm changes the term and document frequencies of the terms in a corpus to a different degree. Hence, the terms have different frequency distributions on the corpus. The frequency fluctuations affect the features and their importance in each experiment because machine learning methods learn from data. Hence, each corpus and its terms are individual according to the applied stemming algorithm. The effect may be positive or negative for retrieval performance. However, when all the features are generalized to the experiments, it is seen that each feature contributes to the selective approach.

IMPLICATIONS

Theoretical implications

In a selective stemming approach, predicting the selection is solely not enough for evaluating IR system performance. To robust IR system, accurately predicting the selections for the given queries that occur high performance differences between participating systems is crucial. Even if the classifier accuracy is moderate, accurately predicting such queries improve retrieval robustness and performance. Binary

classifier has a potential to alleviate the per query performance degradation problem by using the features obtained with term frequency distributions of participating systems. The used classifier appears to achieve this aim by determining useful neighbor queries in decision process. Thereby, the selective approach to stemming minimizes the risk of failure caused by applying stemming to an IR system, and also, according to the experimental results, in most cases, the selective method systematically improves the effectiveness of the systems participating in selection. In most cases, the selective method also systematically improves the effectiveness of the systems participating in selection according to the experimental results. In addition, the proposed work enriches existing literature on selective stemming in terms of improving the robustness and performance of an IR system by predicting to apply stemming to the given query. Also, various features are derived from differences in the term frequency distributions of the systems participating in the selection for this purpose. The contribution of those features to selective stemming is empirically presented.

Practical implications

The proposed selective approach to stemming is convenient to employ an IR system. Because the method works on a document index in which stemming is not applied, and stemming is performed at query time by supplying a conflation set of morphological variants. For instance, document frequencies of the term to be stemmed can be calculated by the posting lists of its morphological variants. Thereby, the classifier features can be calculated at index time or offline, and selective stemming on a query basis is possible to employ in an IR system. Especially for the large-scale Web collections (CW09B and CW12B), the method is reasonably robust against performance fluctuation across queries. The proposed method achieves satisfactory selection, improves the robustness and, in most cases, average retrieval performance of the single system.

CONCLUSIONS

We propose a method for the selective application of stemming on a per query basis in order to alleviate the robustness and effectiveness deteriorations caused by the queries that are harmed by stemming. The proposed selective approach to stemming is a binary classifier to decide whether stemming should be applied to a given query. The classifier uses a rule-based stemmer and pre-retrieval query performance prediction features gathered from the literature. In addition the features in literature, a set of features derived from the frequency distributions of query terms with and without stemming applied is used. The features used in the experiments play a role in different levels of improving retrieval effectiveness according to the query sets. However, on average, features contribute to the retrieval effectiveness of selective stemming in almost equal proportion. Experimental results show that our selective approach is successful at avoiding the risk posed by the queries that are adversely affected by stemming. It is obtained by accurately predicting to apply stemming to the given query. Furthermore, our selective approach is more effective than a single system in which stemming is systematically applied to all queries. This suggests that our proposed selective approach to stemming is both robust and effective. In our following research, we will focus on the classifiers and features to improve the results of this work. We plan to investigate and derive the features that can discriminate queries affected by stemming. Also, we will test advanced machine-learning techniques and pre-trained language models for this purpose.

DECLARATION OF COMPETING INTEREST

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

ACKNOWLEDGMENTS

This work is supported by TÜBİTAK, scientific and technological research projects funding program, under Grant 114E558. Any opinions, findings and conclusions or recommendations expressed in this material are the authors' and do not necessarily reflect those of the sponsor.

APPENDIX

MULTIPLE COMPARISONS

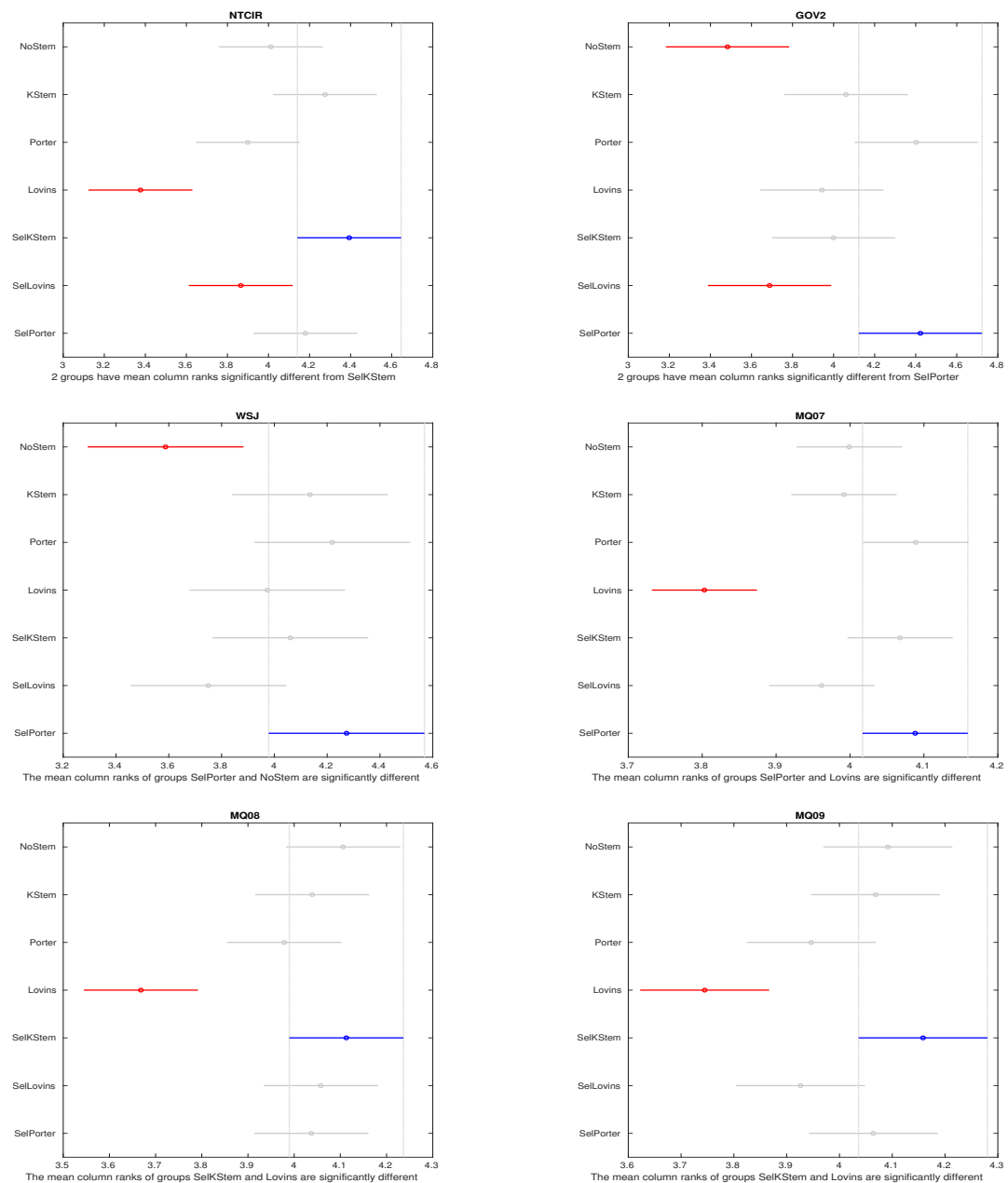



Figure 7. Multiple comparisons based on performance scores of the stemmers in IR systems for NTCIR, GOV2, WSJ, MQ07, MQ08, and MQ09

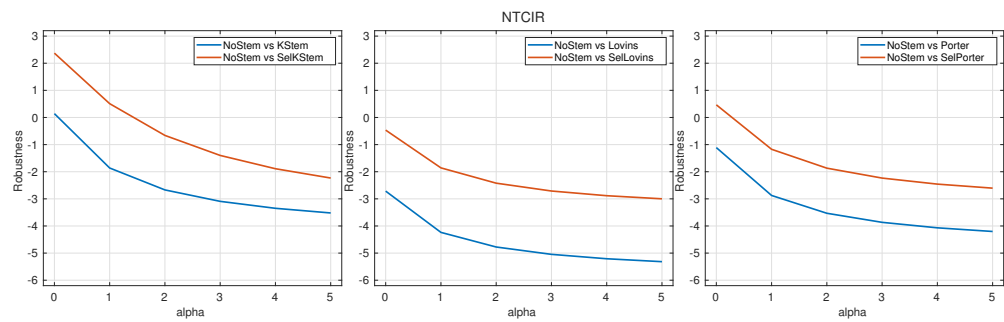
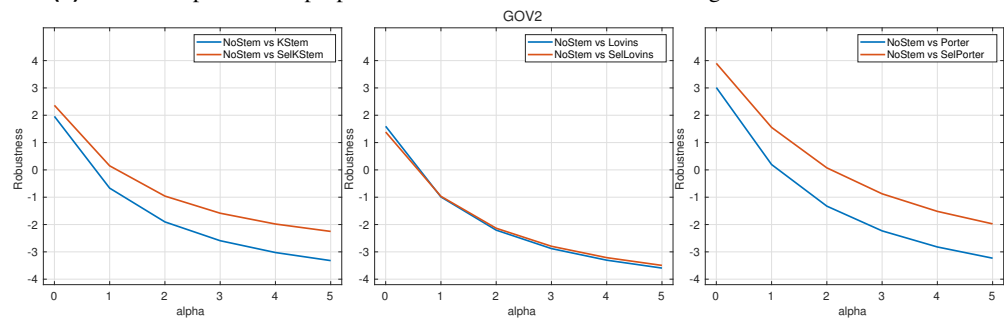
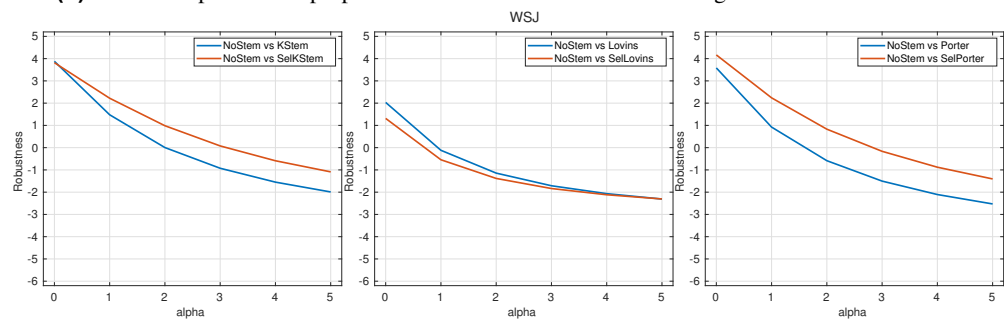
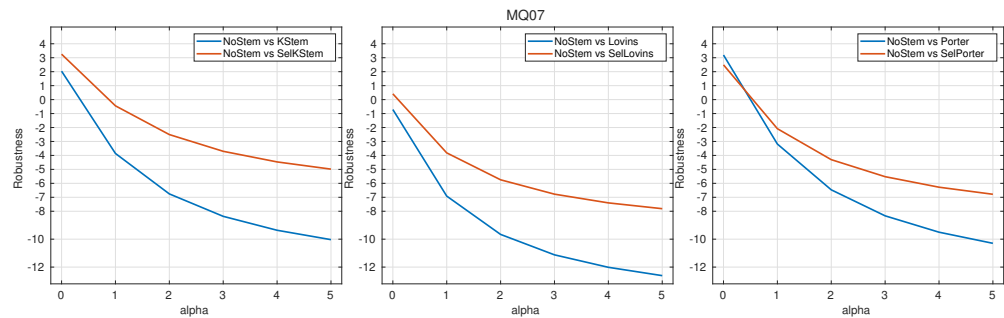
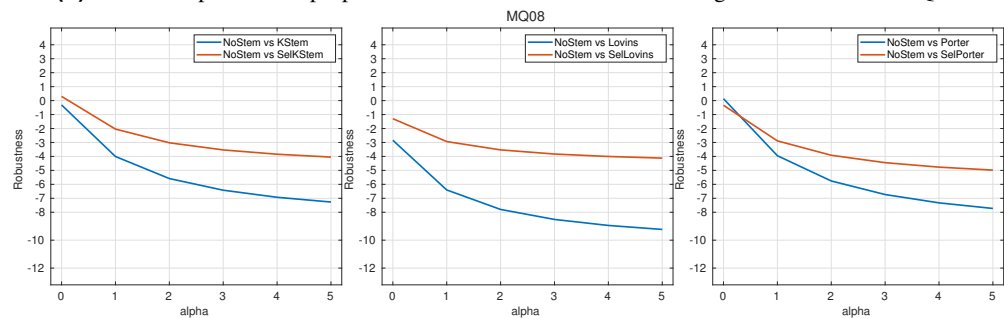
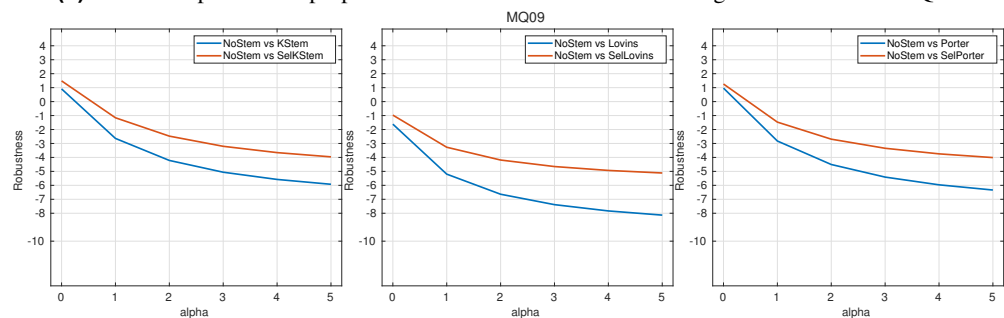
618 **ROBUSTNESS**

(a) TRisk comparisons of proposed method and baseline stemmers against NoStem for NTCIR

(b) TRisk comparisons of proposed method and baseline stemmers against NoStem for GOV2

(c) TRisk comparisons of proposed method and baseline stemmers against NoStem for WSJ

(d) TRisk comparisons of proposed method and baseline stemmers against NoStem for MQ07

(e) TRisk comparisons of proposed method and baseline stemmers against NoStem for MQ08

(f) TRisk comparisons of proposed method and baseline stemmers against NoStem for MQ09

Figure 8. TRisk comparisons of proposed method and baseline stemmers against NoStem for α parameter from 0 to 5.

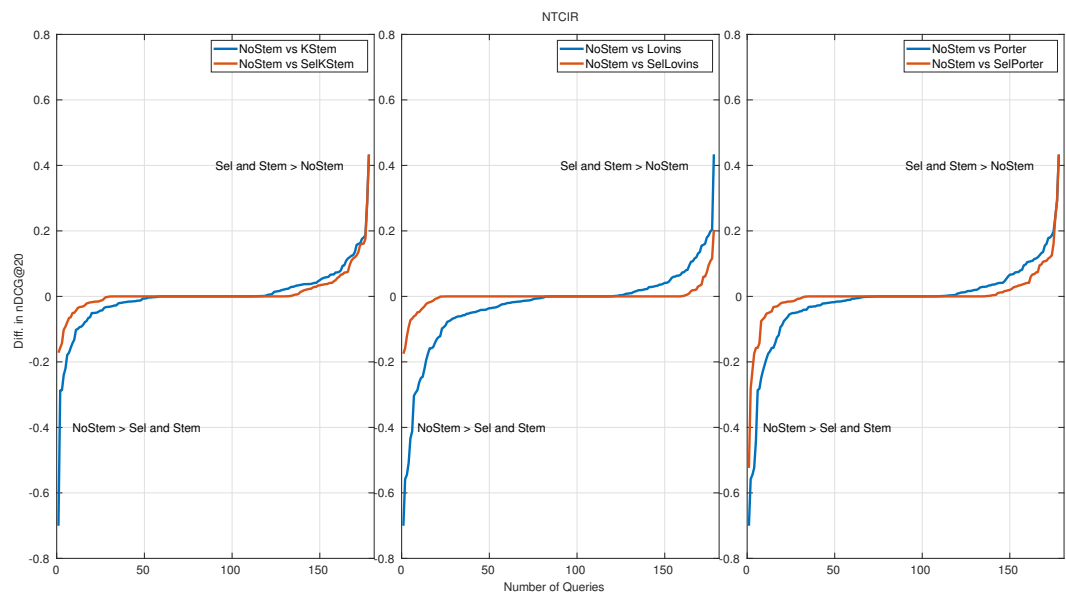
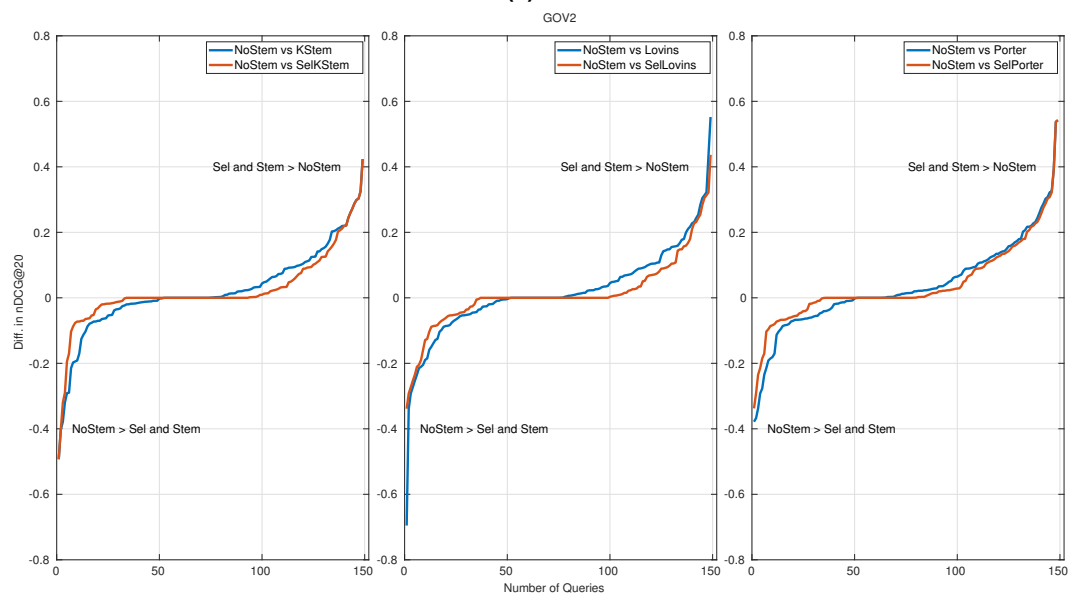
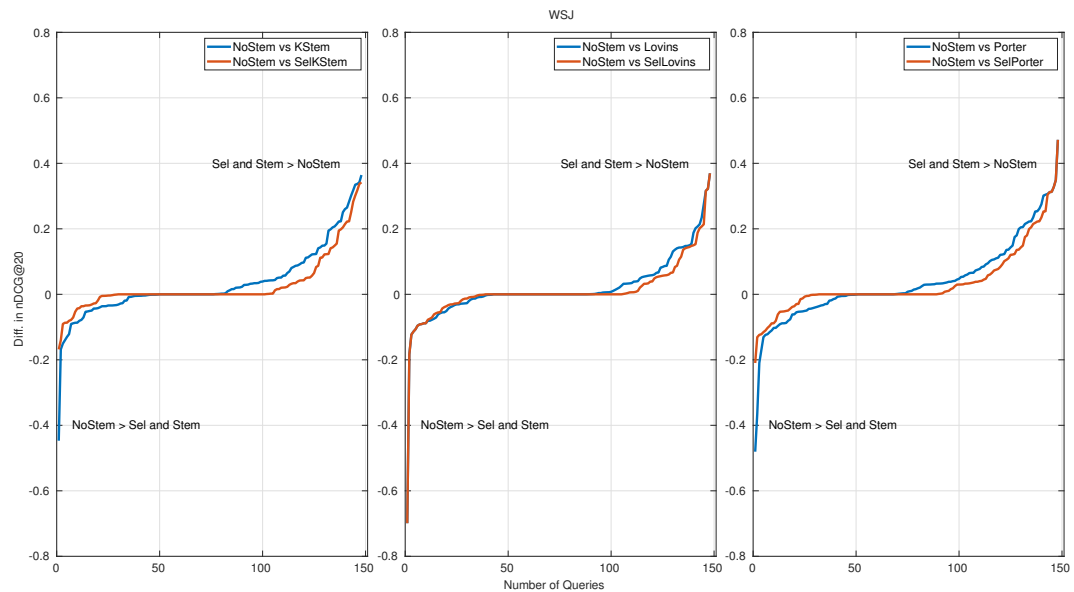
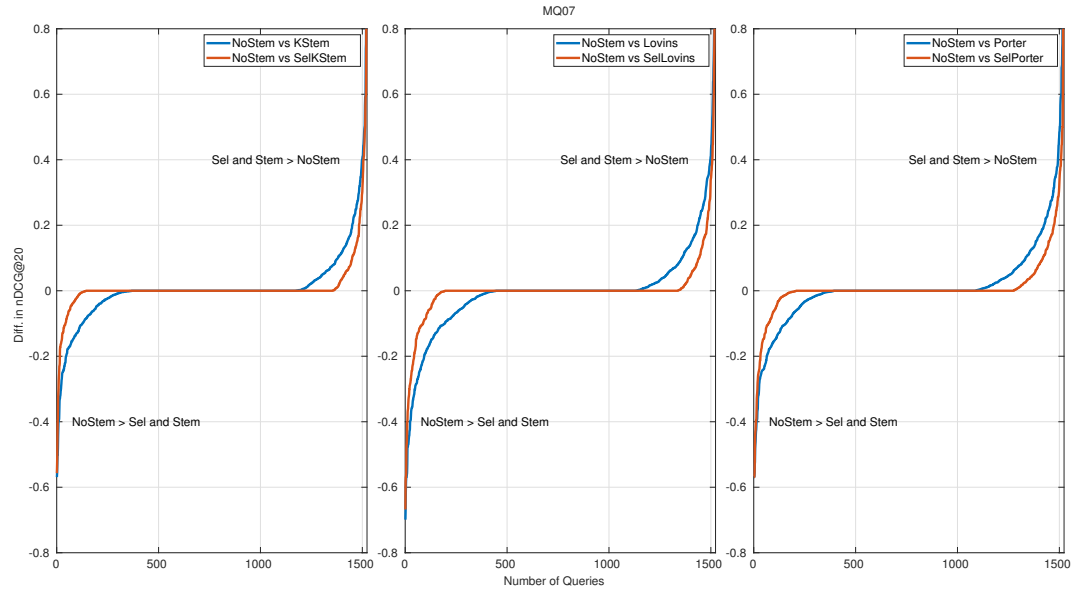
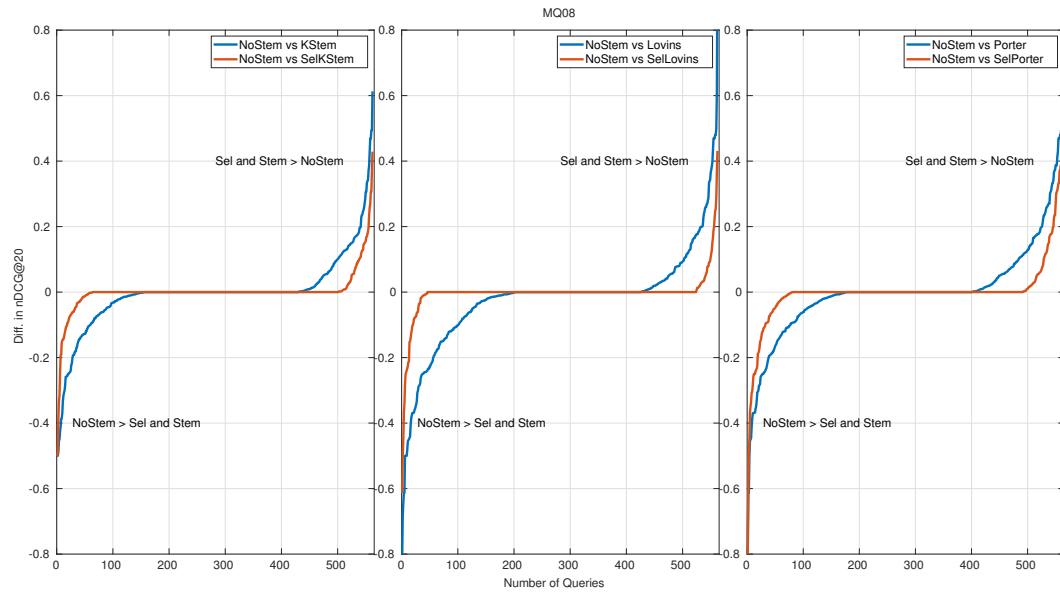
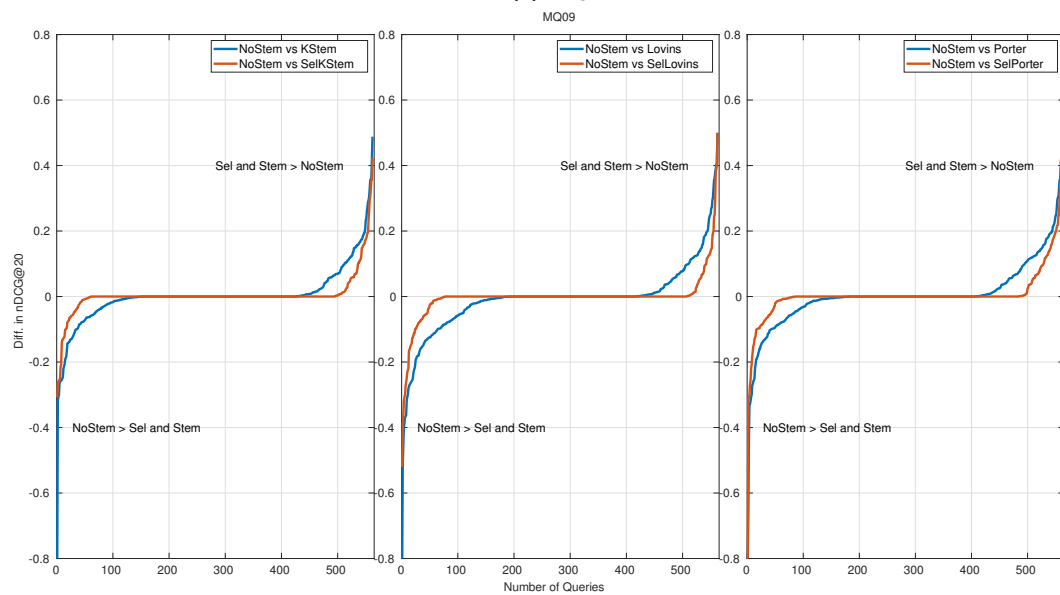
619 **PER QUERY PERFORMANCE DIFFERENCES**

(a) NTCIR

(b) GOV2

(c) WSJ

(d) MQ07

(e) MQ08

(f) MQ09

Figure 9. Per query performance differences

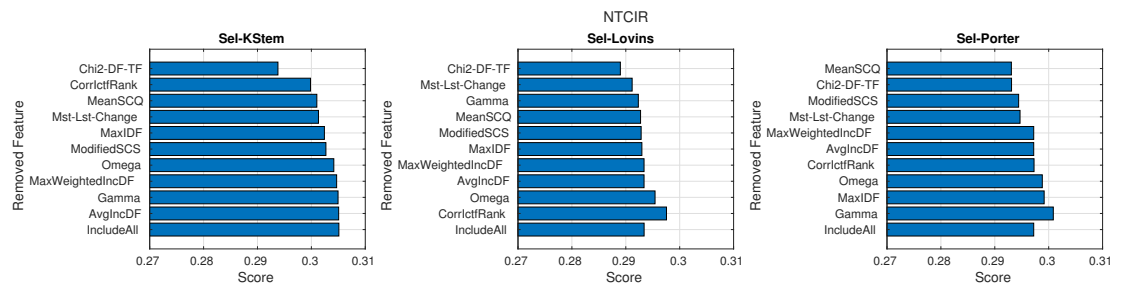
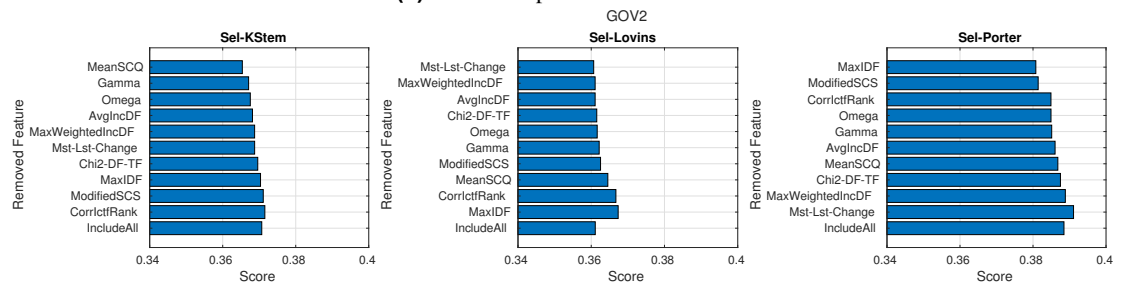
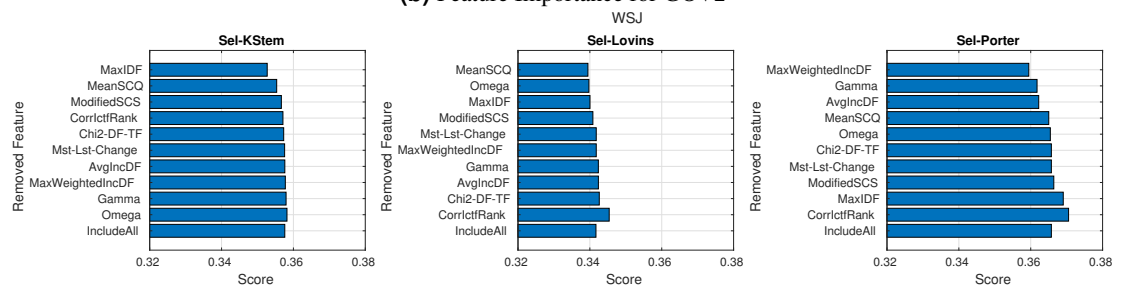
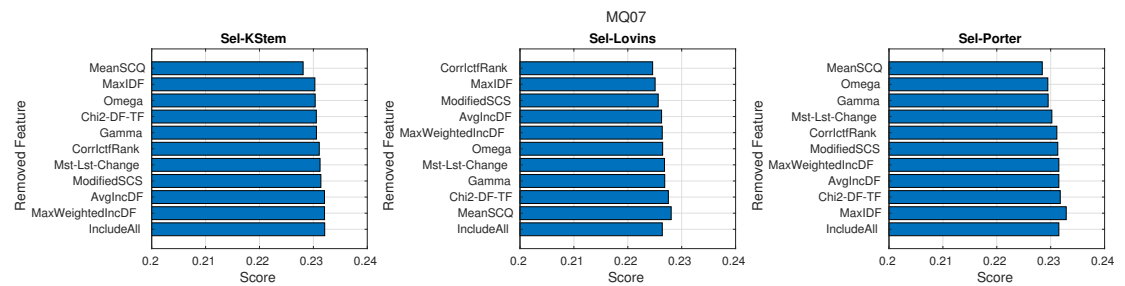
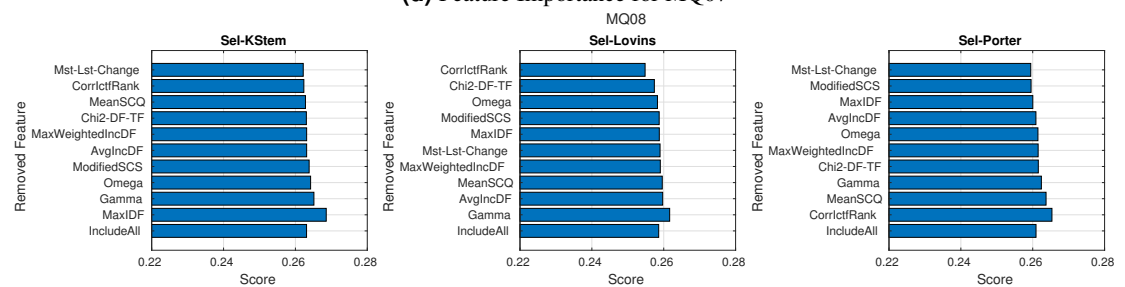
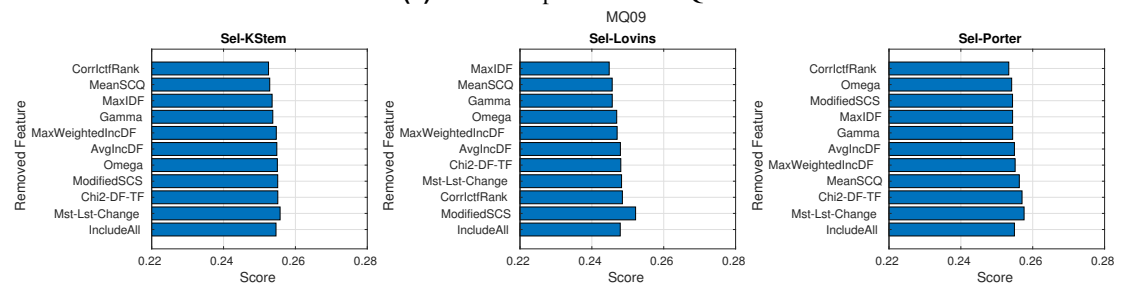
620 **FEATURE IMPORTANCE**

(a) Feature Importance for NTCIR

(b) Feature Importance for GOV2

(c) Feature Importance for WSJ

(d) Feature Importance for MQ07

(e) Feature Importance for MQ08

(f) Feature Importance for MQ09

Figure 10. Feature importance according to the ablation study

REFERENCES

Abuata, B. and Al-Omari, A. (2015). A rule-based stemmer for arabic gulf dialect. *Journal of King Saud University - Computer and Information Sciences*, 27(2):104–112.

Ahmed, F. and Nürnberger, A. (2009). Evaluation of n-gram conflation approaches for arabic text retrieval. *Journal of the American Society for Information Science and Technology*, 60(7):1448–1465.

Al Kharashi, I. A. and Al Sughaiyer, I. A. (2002). Rule merging in a rule-based arabic stemmer. In *Proceedings of the 19th International Conference on Computational Linguistics - Volume 1*, COLING '02, page 1–7, USA. Association for Computational Linguistics.

Alotaibi, F. S. and Gupta, V. (2018). A cognitive inspired unsupervised language-independent text stemmer for information retrieval. *Cognitive Systems Research*, 52:291–300.

Amati, G., Carpineto, C., and Romano, G. (2004). Query difficulty, robustness, and selective application of query expansion. In McDonald, S. and Tait, J., editors, *Advances in Information Retrieval, 26th European Conference on IR Research, ECIR 2004, Sunderland, UK, April 5-7, 2004, Proceedings*, volume 2997 of *Lecture Notes in Computer Science*, pages 127–137, Berlin, Heidelberg. Springer Berlin Heidelberg.

Arguello, J., Crane, M., Diaz, F., Lin, J., and Trotman, A. (2016). Report on the SIGIR 2015 workshop on reproducibility, inexplicability, and generalizability of results (RIGOR). *SIGIR Forum*, 49(2):107–116.

Arlot, S. and Celisse, A. (2010). A survey of cross-validation procedures for model selection. *Statistics Surveys*, 4:40–79.

Arslan, A. and Dinçer, B. T. (2019). A selective approach to index term weighting for robust information retrieval based on the frequency distributions of query terms. *Information Retrieval Journal*, 22(6):543–569.

Azzopardi, L., Crane, M., Fang, H., Ingersoll, G., Lin, J., Moshfeghi, Y., Scells, H., Yang, P., and Zuccon, G. (2017). The Lucene for information access and retrieval research (LIARR) workshop at SIGIR 2017. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '17, pages 1429–1430, Shinjuku, Tokyo, Japan.

Balasubramanian, N. and Allan, J. (2010). Learning to select rankers. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '10, pages 855–856, Geneva, Switzerland. Association for Computing Machinery.

Basu, M., Roy, A., Ghosh, K., Bandyopadhyay, S., and Ghosh, S. (2017). A novel word embedding based stemming approach for microblog retrieval during disasters. In Jose, J. M., Hauff, C., Altıngövde, İ. Ş., Song, D., Albakour, D., Watt, S., and Tait, J., editors, *Advances in Information Retrieval - 39th European Conference on IR Research, ECIR 2017, Aberdeen, UK, April 8-13, 2017, Proceedings*, volume 10193 of *Lecture Notes in Computer Science*, pages 589–597, Cham. Springer International Publishing.

Białecki, A., Muir, R., and Ingersoll, G. (2012). Apache Lucene 4. In *Proceedings of the SIGIR 2012 Workshop on Open Source Information Retrieval*, pages 17–24, Portland, Oregon, USA.

Bigot, A., Dejean, S., and Mothe, J. (2015). Learning to Choose the Best System Configuration in Information Retrieval: the case of repeated queries. *Journal of Universal Computer Science*, 21(13):1726–1745.

Bölücü, N. and Can, B. (2019). Unsupervised joint pos tagging and stemming for agglutinative languages. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, 18(3).

Brychcín, T. and Konopík, M. (2015). Hps: High precision stemmer. *Information Processing & Management*, 51(1):68–91.

Buckley, C. (2004). Why current IR engines fail. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '04, pages 584–585, Sheffield, United Kingdom.

Buckley, C. (2009). Why current IR engines fail. *Information Retrieval*, 12(6):652–665.

Cao, G., Nie, J.-Y., Gao, J., and Robertson, S. (2008a). Selecting good expansion terms for pseudo-relevance feedback. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '08, pages 243–250, Singapore, Singapore. Association for Computing Machinery.

Cao, G., Robertson, S., and Nie, J.-Y. (2008b). Selecting query term alternations for web search by exploiting query contexts. In *Proceedings of ACL-08: HLT*, pages 148–155, Columbus, Ohio, USA.

Carmel, D. and Yom-Tov, E. (2010). *Estimating the query difficulty for information retrieval*. Morgan &

676   Claypool Publishers.

677 Chin, S.-C., DeCook, R., Street, W. N., and Eichmann, D. (2010). Query-based text normalization
678   selection models for enhanced retrieval accuracy. In *Proceedings of the NAACL HLT 2010 Workshop*
679   *on Semantic Search*, SS '10, pages 19–26, Los Angeles, California. Association for Computational
680   Linguistics.

681 Church, K. and Gale, W. (1999). *Inverse Document Frequency (IDF): A Measure of Deviations from*
682   *Poisson*, pages 283–295. Springer Netherlands, Dordrecht.

683 Clarke, C. L., Craswell, N., and Soboroff, I. (2004). Overview of the TREC 2004 terabyte track. In
684   *Proceedings of the 2004 Text Retrieval Conference*, NIST Special Publication. National Institute of
685   Standards and Technology (NIST).

686 Croft, B. and Xu, J. (1995). Corpus-specific stemming using word form co-occurence. In *In Fourth*
687   *Annual Symposium on Document Analysis and Information Retrieval*, pages 147–159, Las Vegas,
688   Nevada, USA.

689 Cronen-Townsend, S., Zhou, Y., and Croft, W. B. (2002). Predicting query performance. In *Proceedings*
690   *of the 25th Annual International ACM SIGIR Conference on Research and Development in Information*
691   *Retrieval*, SIGIR '02, page 299–306. Association for Computing Machinery.

692 Cronen-Townsend, S., Zhou, Y., and Croft, W. B. (2004). A framework for selective query expansion.
693   In *Proceedings of the Thirteenth ACM International Conference on Information and Knowledge*
694   *Management*, CIKM '04, pages 236–237, Washington, D.C., USA. Association for Computing
695   Machinery.

696 Deveaud, R., Mothe, J., Ullah, M. Z., and Nie, J.-Y. (2018). Learning to adaptively rank document
697   retrieval system configurations. *ACM Trans. Inf. Syst.*, 37(1).

698 Dinçer, B. T., Macdonald, C., and Ounis, I. (2014). Hypothesis testing for the risk-sensitive evaluation of
699   retrieval systems. In *Proceedings of the 37th International ACM SIGIR Conference on Research and*
700   *Development in Information Retrieval*, SIGIR '14, pages 23–32, Gold Coast, Queensland, Australia.

701 Ghanbari, E. and Shakery, A. (2019). Query-dependent learning to rank for cross-lingual information
702   retrieval. *Knowledge and Information Systems*, 59(3):711–743.

703 Goldsmith, J. (2001). Unsupervised Learning of the Morphology of a Natural Language. *Computational*
704   *Linguistics*, 27(2):153–198.

705 Gupta, V., Joshi, N., and Mathur, I. (2013). Rule based stemmer in urdu. In *2013 4th International*
706   *Conference on Computer and Communication Technology (ICCCT)*, pages 129–132. IEEE.

707 Harman, D. (1987). A failure analysis of the limitation of suffixing in an online environment. In
708   *Proceedings of the 10th Annual International ACM SIGIR Conference on Research and Development*
709   *in Information Retrieval*, SIGIR '87, pages 102–107, New Orleans, Louisiana, USA.

710 Harman, D. (1991). How effective is suffixing? *Journal of the American Society for Information Science*,
711   42(1):7–15.

712 Harman, D. and Buckley, C. (2004). The NRRC reliable information access (RIA) workshop. In
713   *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development*
714   *in Information Retrieval*, SIGIR '04, pages 528–529, Sheffield, United Kingdom.

715 Hauff, C., Azzopardi, L., Hiemstra, D., and de Jong, F. (2010). Query performance prediction: Evaluation
716   contrasted with effectiveness. In Gurrin, C., He, Y., Kazai, G., Kruschwitz, U., Little, S., Roelleke,
717   T., Rüger, S., and van Rijsbergen, K., editors, *Advances in Information Retrieval - 32nd European*
718   *Conference on IR Research, ECIR 2010, Milton Keynes, UK, March 28-31, 2010.Proceedings*, volume
719   5993 of *Lecture Notes in Computer Science*, pages 204–216, Berlin, Heidelberg. Springer Berlin
720   Heidelberg.

721 He, B. and Ounis, I. (2003). University of glasgow at the robust track- A query-based model selection
722   approach for the poorly-performing queries. In Voorhees, E. M. and Buckland, L. P., editors,
723   *Proceedings of The Twelfth Text REtrieval Conference, TREC 2003, Gaithersburg, Maryland, USA,*
724   *November 18-21, 2003*, volume 500-255 of *NIST Special Publication*, pages 636–645. National Institute
725   of Standards and Technology (NIST).

726 He, B. and Ounis, I. (2004a). Inferring query performance using pre-retrieval predictors. In *String*
727   *Processing and Information Retrieval - 11th International Conference, SPIRE 2004, Padova, Italy,*
728   *October 5-8, 2004. Proceedings*, volume 3246 of *Lecture Notes in Computer Science*, pages 43–54.
729   Springer Berlin Heidelberg.

730 He, B. and Ounis, I. (2004b). A query-based pre-retrieval model selection approach to information

731   retrieval. In *Proceedings of RIAO 2004 - Coupling Approaches, Coupling Media and Coupling*
732   *Languages for Information Retrieval*, RIAO '04, pages 706–719, Vaucluse, France. LE CENTRE DE
733   HAUTES ETUDES INTERNATIONALES D'INFORMATIQUE DOCUMENTAIRE.

734 Hull, D. A. (1996). Stemming algorithms: A case study for detailed evaluation. *Journal of the American*
735   *Society for Information Science*, 47(1):70–84.

736 Järvelin, K. and Kekäläinen, J. (2000). IR evaluation methods for retrieving highly relevant documents. In
737   *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development*
738   *in Information Retrieval*, SIGIR '00, pages 41–48, Athens, Greece.

739 Järvelin, K. and Kekäläinen, J. (2002). Cumulated gain-based evaluation of IR techniques. *ACM Trans.*
740   *Inf. Syst.*, 20(4):422–446.

741 Kim, Y., Callan, J., Culpepper, J. S., and Moffat, A. (2016). Load-balancing in distributed selective search.
742   In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in*
743   *Information Retrieval*, SIGIR '16, pages 905–908, Pisa, Italy. Association for Computing Machinery.

744 Kim, Y., Callan, J., Culpepper, J. S., and Moffat, A. (2017). Efficient distributed selective search.
745   *Information Retrieval Journal*, 20(3):221–252.

746 Krovetz, R. (1993). Viewing morphology as an inference process. In *Proceedings of the 16th Annual*
747   *International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR
748   '93, pages 191–202, Pittsburgh, Pennsylvania, USA. ACM.

749 Kulkarni, A. and Callan, J. (2015). Selective search: Efficient and effective search of large textual
750   collections. *ACM Trans. Inf. Syst.*, 33(4):17:1–17:33.

751 Lin, J., Crane, M., Trotman, A., Callan, J., Chattopadhyaya, I., Foley, J., Ingersoll, G., Macdonald, C.,
752   and Vigna, S. (2016). Toward reproducible baselines: The open-source IR reproducibility challenge. In
753   *Advances in Information Retrieval: 38th European Conference on IR Research, ECIR 2016, Padua,*
754   *Italy, March 20-23, 2016. Proceedings*, volume 9626 of *Lecture Notes in Computer Science*, pages
755   408–420. Springer International Publishing, Cham.

756 Ljubešić, N., Boras, D., and Kubelka, O. (2007). Retrieving information in croatian: Building a simple and
757   efficient rule-based stemmer. In *Proceedings of the Conference on Digital Information and Heritage.*
758   *Zagreb*, INFuture2007, pages 313–320. Odsjek za informacijske znanosti, Filozofski fakultet, Zagreb.

759 Lovins, J. B. (1968). Development of a stemming algorithm. *Mechanical Translation and Computational*
760   *Linguistics*, 11(1-2):22–31.

761 Luo, C., Sakai, T., Liu, Y., Dou, Z., Xiong, C., and Xu, J. (2017). Overview of the NTCIR-13 we want web
762   task. *Proceedings of the 13th NTCIR Conference on Evaluation of Information Access Technologies*,
763   pages 394–401.

764 Mahmud, M. R., Afrin, M., Razzaque, M. A., Miller, E., and Iwashige, J. (2014). A rule based
765   bengali stemmer. In *2014 International Conference on Advances in Computing, Communications and*
766   *Informatics (ICACCI)*, pages 2750–2756. IEEE.

767 Mao, J., Sakai, T., Luo, C., Xiao, P., Liu, Y., and Dou, Z. (2019). Overview of the NTCIR-14 we want web
768   task. In *Proceedings of the 14th NTCIR Conference on Evaluation of Information Access Technologies*,
769   pages 455–467.

770 McNamee, P. and Mayfield, J. (2004). Character n-gram tokenization for european language text retrieval.
771   *Information Retrieval*, 7(1):73–97.

772 Mothe, J. and Ullah, M. Z. (2021). Defining an optimal configuration set for selective search strategy -
773   a risk-sensitive approach. In *Proceedings of the 30th ACM International Conference on Information*
774   *& Knowledge Management*, CIKM '21, pages 1335–1345, Virtual Event, Queensland, Australia.
775   Association for Computing Machinery.

776 Oard, D. W., Levow, G.-A., and Cabezas, C. I. (2001). Clef experiments at maryland: Statistical stemming
777   and backoff translation. In Peters, C., editor, *Cross-Language Information Retrieval and Evaluation -*
778   *Workshop of the Cross-Language Evaluation Forum, CLEF 2000 Lisbon, Portugal, September 21–22,*
779   *2000 Revised Papers*, volume 2069 of *Lecture Notes in Computer Science*, pages 176–187, Berlin,
780   Heidelberg. Springer Berlin Heidelberg.

781 Paice, C. D. (1990). Another stemmer. *SIGIR Forum*, 24(3):56–61.

782 Paik, J. H., Mitra, M., Parui, S. K., and Järvelin, K. (2011a). Gras: An effective and efficient stemming
783   algorithm for information retrieval. *ACM Trans. Inf. Syst.*, 29(4).

784 Paik, J. H., Pal, D., and Parui, S. K. (2011b). A novel corpus-based stemming algorithm using co-
785   occurrence statistics. In *Proceedings of the 34th International ACM SIGIR Conference on Research*

and Development in Information Retrieval, SIGIR '11, pages 863–872, Beijing, China. Association for Computing Machinery.

Paik, J. H., Parui, S. K., Pal, D., and Robertson, S. E. (2013). Effective and robust query-based stemming. *ACM Trans. Inf. Syst.*, 31(4).

Pande, B. P., Tamta, P., and Dhami, H. S. (2018). Generation, implementation, and appraisal of an N-gram-based stemming algorithm. *Digital Scholarship in the Humanities*, 34(3):558–568.

Patil, H. B. and Patil, A. S. (2017). Mars: A rule-based stemmer for morphologically rich language marathi. In *2017 International Conference on Computer, Communications and Electronics (Comptelix)*, pages 580–584. IEEE.

Peng, F., Ahmed, N., Li, X., and Lu, Y. (2007). Context sensitive stemming for web search. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '07, pages 639–646, Amsterdam, The Netherlands.

Peng, J., Macdonald, C., and Ounis, I. (2010). Learning to select a ranking function. In Gurrin, C., He, Y., Kazai, G., Kruschwitz, U., Little, S., Roelleke, T., Rüger, S., and van Rijsbergen, K., editors, *Advances in Information Retrieval - 32nd European Conference on IR Research, ECIR 2010, Milton Keynes, UK, March 28-31, 2010.Proceedings*, volume 5993 of *Lecture Notes in Computer Science*, pages 114–126, Berlin, Heidelberg. Springer Berlin Heidelberg.

Porter, M. F. (1980). An algorithm for suffix stripping. *Program*, 14(3):130–137.

Porter, M. F. (1997). An algorithm for suffix stripping. In *Readings in Information Retrieval*, pages 313–316. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Robertson, S. (2004). Understanding inverse document frequency: on theoretical arguments for IDF. *Journal of Documentation*, 60(5):503–520.

Robertson, S. and Zaragoza, H. (2009). The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389.

Roy, A., Ghorai, T., Ghosh, K., and Ghosh, S. (2017). Combining local and global word embeddings for microblog stemming. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, CIKM '17, pages 2267–2270, Singapore, Singapore. Association for Computing Machinery.

Saleh, S. and Pecina, P. (2019). Term selection for query expansion in medical cross-lingual information retrieval. In Azzopardi, L., Stein, B., Fuhr, N., Mayr, P., Hauff, C., and Hiemstra, D., editors, *Advances in Information Retrieval - 41st European Conference on IR Research, ECIR 2019, Cologne, Germany, April 14–18, 2019, Proceedings, Part I*, volume 11437 of *Lecture Notes in Computer Science*, pages 507–522, Cham. Springer International Publishing.

Sarkar, S. and Bandyopadhyay, S. (2008). Design of a rule-based stemmer for natural language text in bengali. In *Proceedings of the IJCNLP-08 workshop on NLP for Less Privileged Languages*, page 65–72.

Singh, J. and Gupta, V. (2019). A novel unsupervised corpus-based stemming technique using lexicon and corpus statistics. *Knowledge-Based Systems*, 180:147–162.

Singh, P. and Bhowmick, P. K. (2022). Neural network guided fast and efficient query-based stemming by predicting term co-occurrence statistics. *SN Computer Science*, 3(3):198.

Spärck Jones, K. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28(1):11–21.

Tonellotto, N., Macdonald, C., and Ounis, I. (2013). Efficient and effective retrieval using selective pruning. In *Proceedings of the Sixth ACM International Conference on Web Search and Data Mining*, WSDM '13, pages 63–72, Rome, Italy. Association for Computing Machinery.

Voorhees, E. M., Rajput, S., and Soboroff, I. (2016). Promoting repeatability through open runs. In *Proceedings of the Seventh International Workshop on Evaluating Information Access*, EVIA 2016, pages 17–20, Tokyo, Japan.

Wood, V. (2013). Improving query term expansion with machine learning. Master's thesis, University of Otago.

Zhao, Y., Scholer, F., and Tsegay, Y. (2008). Effective pre-retrieval query performance prediction using similarity and variability evidence. In *Advances in Information Retrieval*, volume 4956 of *Lecture Notes in Computer Science*, pages 52–64. Springer Berlin Heidelberg.