Query-based Text Normalization Selection Models for Enhanced Retrieval Accuracy

Si-Chi Chin Rhonda DeCook W. Nick Street David Eichmann
The University of Iowa
Iowa City, USA.
{si-chi-chin, rhonda-decook, nick-street, david-eichmann}@uiowa.edu

Abstract

Text normalization transforms words into a base form so that terms from common equivalent classes match. Traditionally, information retrieval systems employ stemming techniques to remove derivational affixes. Depluralization, the transformation of plurals into singular forms, is also used as a low-level text normalization technique to preserve more precise lexical semantics of text.

Experiment results suggest that the choice of text normalization technique should be made individually on each topic to enhance information retrieval accuracy. This paper proposes a hybrid approach, constructing a query-based selection model to select the appropriate text normalization technique (stemming, depluralization, or not doing any text normalization). The selection model utilized ambiguity properties extracted from queries to train a composite of Support Vector Regression (SVR) models to predict a text normalization technique that yields the highest Mean Average Precision (MAP). Based on our study, such a selection model holds promise in improving retrieval accuracy.

1 Introduction

Stemming removes derivational affixes of terms therefore allowing terms from common equivalence classes to be clustered. However, stemming also introduces noise by mapping words of different concepts or meanings into one base form, thus impeding word-sense disambiguation. Depluralization, the conversion of plural word forms to singular form, preserves more precise semantics of text than stemming (Krovetz, 2000).

Empirical research has demonstrated the ambivalent effect of stemming on text retrieval performance. Hull (1996) conducted a comprehensive case study on the effects of four stemmer techniques and the removal of plural "s" [1] on retrieval performance. Hull suggested that the adoption of stemming is beneficial but plural removal is as well competitive when the size of documents is small. Prior research (Manning and Schtze, 1999; McNamee et al., 2008) indicated that traditional stemming, though still benefiting some queries, would not necessarily enhance the average retrieval performance. In addition, stemming was considered one of the technique failures undermining retrieval performance in the TREC 2004 Robust Track (Voorhees, 2006). Prior research also noted the semantic differences between plurals and singulars. Riloff (1995) indicated that plural and singular nouns are distinct because plural nouns usually pertain to the "general types of incidents," while singular nouns often pertain to "a specific incident."

Nevertheless, prior research has not closely examined the effect of the change of the semantics caused by different level of text normalization techniques. In our work, we conducted extensive experiments on the TREC 2004 Robust track collection to evaluate the effect of stemming and depluralization on information retrieval. In addition, we quantify the ambiguity of a query, extracting five ambiguity properties from queries. The extracted ambiguity properties are used to construct query-based selection model, a composite of multiple Support Vector

[1] In our work, we not only removed the plural "s" or "es" but also changed irregular plural forms such as "children" to its singular form "child".

Regression models, to determine the most appropriate text normalization technique for a given query. To our knowledge, our work is the first study to construct a query-based selection model, using ambiguity properties extracted from provided queries to select an optimal text normalization technique for each query.

The remainder of this paper is organized as follows. In section 2 we describe our experimental setups and dataset. Section 3 describes and analyzes experiment results of different text normalization techniques on the dataset. We discuss five ambiguity properties and validate each property in section 4. In section 5 we describe the framework and the prediction results of the proposed query-based selection model. Finally, we summarize our findings and discuss future work in section 6.

2 Experiment Setup

The experiment utilizes the queries and relevance judgment results from the TREC 2004 Robust Track to evaluate the effect of three text normalization techniques – raw text, depluralized text, and stemmed text. The TREC 2004 Robust Track used a document set of approximately 528,000 documents comprising 1,904 MB of text. In total, 249 query topics were used in TREC Robust 2004.

Figure 1 illustrates the setup of the experiment. The collection is parsed with a SAX parser and stored in a Postgres database. Lucene is then used to generate three indices: indices of raw text, depluralized text, and stemmed text. The Postgres database stores each document of the collection, the query topics of the TREC 2004 Robust Track, and results of experiments. The ambiguity properties for each query is also computed in the Postgres system. We query Lucene indices to obtain the top 1,000 relevant results and compute Mean Average Precision (MAP) with the trec_eval program to evaluate performance. We use R to analyze performance scores, generate descriptive charts, conduct non-parametric statistical tests, and perform a paired t-test. We use Weka (Hall et al., 2009) to construct query-based selection model that incorporates multiple Support Vector Regression (SVR) models.

2.1 Query Models

The TREC 2004 Robust Track provides 249 query topics; each includes a title, a short description, and a narrative (usually one-paragraph). We selected three basic query models as a modest baseline to demonstrate the effect of different text normalization techniques. Our future work will exploit other ranking models such as BM25 and LMIR. The three query models used in the experiment are: (1) boolean search with the title words of topics concatenated with logical AND (e.g. hydrogen AND fuel AND automobiles); (2) boolean search with the title words of topics concatenated with logical OR (e.g. hydrogen OR fuel OR automobiles); (3) cosine similarity with the title words of topics. Lucene MoreLikeThis (MLT) class supports both boolean and cosine similarity query methods for the experiment. Figure 2 shows how query topics are processed before interrogating the indices. Original queries are first depluralized or stemmed, further processed according to each query model, and finally run against the depluralized and stemmed indices. The experiment runs unprocessed raw queries against the index of raw text, depluralized queries against the index of depluralized text, and stemmed queries against the index of stemmed text.

3 Experiment Results

Table 3 and Figure 3 summarize the results for the full set of topics. Each row in Table 3 represents a query model combined with a given text normalization technique as described in section 2.1.

For each query model and text normalization technique, we present the MAP value computed across all relevant topics. We also provide the p-value for comparing MAP between each normalization technique and the baseline (i.e. non-normalized (raw) queries). The p-value is generated from the pairwise Wilcoxon signed rank sum test. Figure 3 describes the distribution of MAP across the three text normalization techniques and three query models. The distributions are skewed and many outliers are observed. In general, boolean OR and MLT query models perform similarly and stemming has the highest median MAP value across all three query models. The results from Table 3 for the combined topic set show that depluralization and stemming perform significantly better than the raw baseline. However, the performance difference between depluralization and stemming is not significant except for the AND boolean query model. In general, the differences of MAP among three text normalization

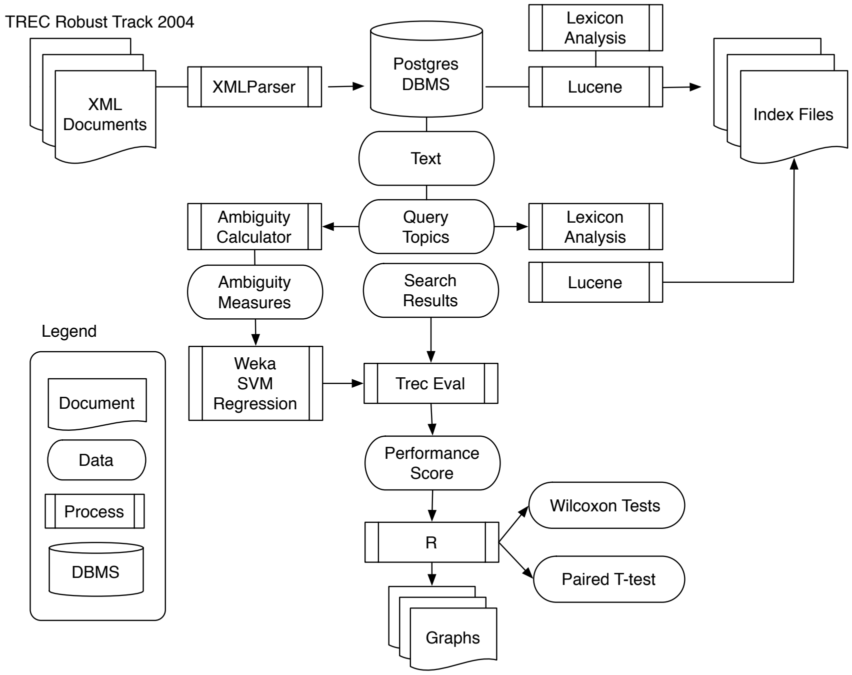

Figure 1: Flow chart of experiment setup

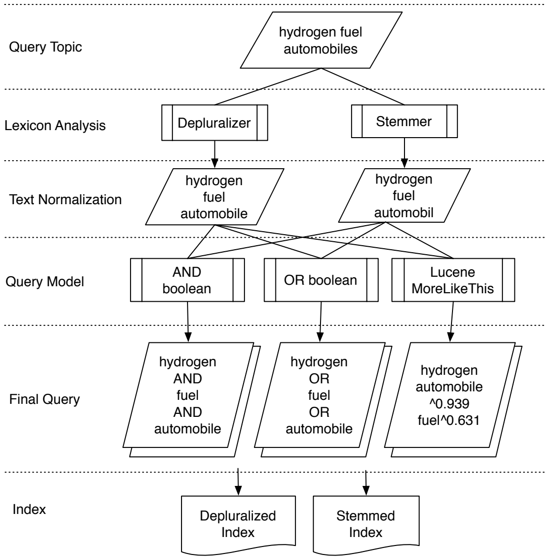

Figure 2: Using the query "hydrogen fuel automobiles" as an example, the depluralized query becomes "hydrogen fuel automobile" and the stemmed query becomes "hydrogen fuel automobil." Final boolean queries for depluralized topic become "hydrogen AND fuel AND automobile" and "hydrogen OR fuel OR automobil." MoreLikeThis (MLT) is the Lucene class used for cosine similarity retrieval. A term vector score appends each word in the topic.

techniques are within 2%.

To visualize the relative performances among three text normalization techniques, we standardized the three MAP values for a single topic (one from each text normalization technique) to have mean 0 and standard deviation of 1. The result provides a 3-value pattern emphasizing the ordering of the MAPs across the text normalization techniques, rather than the raw MAP values themselves. We then used the K-medoids algorithm (Kaufman and Rousseeuw, 1990) to cluster the transformed data, applying Euclidean distance as the distance measure. Figure 4 is an example of 9 constructed clusters based on the MAP scores of the MLT query model. In a cluster, a line represents the standardized MAP value of a topic on each text normalization technique. Given the small differences in aggregate MAP performance, it is interesting to note that the clusters demonstrate variable patterns, indicating that some topics performed better as a depluralized query than a stemmed query.

The cluster analysis suggests that the choice of text normalization technique should be made individually on each topic. As we choose an appropriate text normalization technique for a given topic, we would further enhance retrieval performance. In

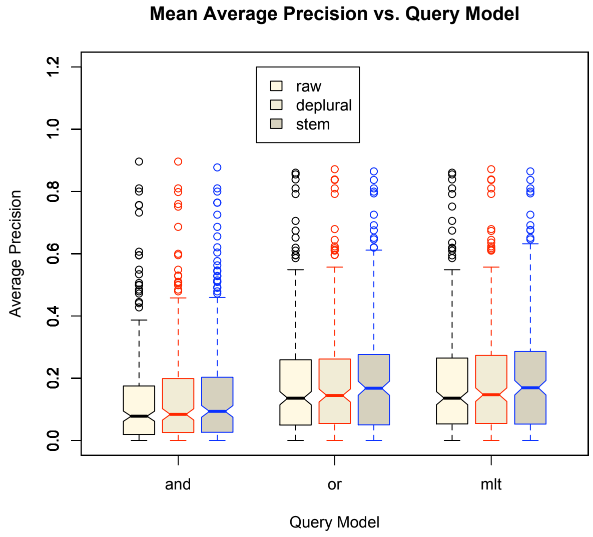

Figure 3: Profile plot of MAP

the next section, we address the issue of inconsistent performance by constructing regression models to predict the mean average precision of each query from the ambiguity measures, and choose an appropriate normalization method based on these predictions.

4 Ambiguity Properties

Research has affirmed the negative impact of query ambiguity on an information retrieval system. As stemming clusters terms of different concepts, it should increase query ambiguity. To quantify the query ambiguity potentially caused by stemming, we compute five ambiguity properties for each query: 1) the product of the number of senses, referred as the sense product; 2) the product of the number of words mapped to one base form (e.g. a stem), referred as the word product; 3) the ratio of the sense product of depluralized query to which of stemmed query, referred as the deplural-stem ratio; 4) the sum of the inverse document frequency for each word in a query, referred as the idf-sum; 5) the length of a query.

4.1 Sense Product

Sense product measures the extent of query ambiguity after stemming. We first find all words mapped to a given stem and, for each word, we then count the number of senses found in WordNet. To compute

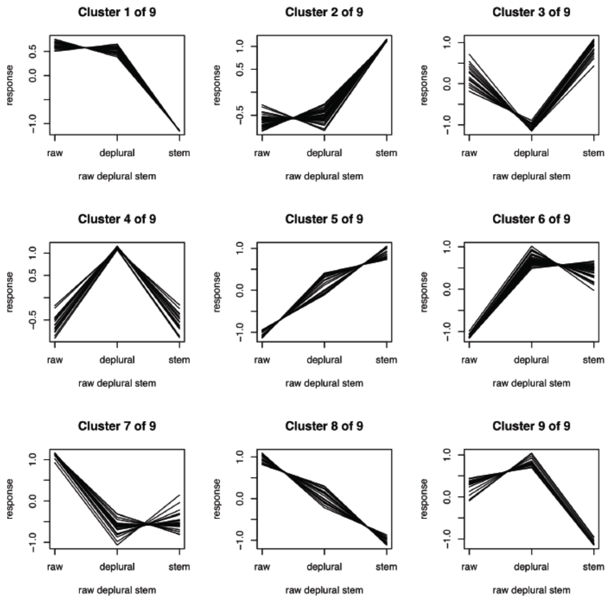

Figure 4: Example of relative performance similarities among text normalization techniques. The cluster analysis uses the MAP scores of the MLT query model

	Combined Topic Set	
Run	MAP	p-value
AND_Raw	0.1213	N/A
AND_Dep	0.1324	5.598e-06*
AND_Stem	0.1550 †	1.599e-07*
OR_Raw	0.1851	N/A
OR_Dep	0.1922	0.03035*
OR_Stem	0.2069	0.01123*
MLT_Raw	0.1893	N/A
MLT_Dep	0.1959	0.04837*
MLT_Stem	0.2093	0.009955*

Table 1: Paired Wilcoxon signed-ranked test on Mean Average Precision (MAP), utilizing raw query as the baseline. Significant differences between query models are labeled with *. Results labeled with † indicate significant differences between depluralized queries and a stemmed queries.

the number of senses for a given stem, we accumulate the number of senses of each word mapped to the stem. The sense product is then the multiplying of the number of senses for each stemmed query term, computed as:

$$\textit{sense\_product} = \prod_{i=1}^{n} \sum_{j=1}^{m} S_j \qquad (1)$$

S_j denotes the number of senses for each word mapped to a stem i. We have m words mapped to a stem i and have n stems in a query. As the sense product increases, the query ambiguity increases. Figure 5 illustrates the computation of the sense product for the query "organic soil enhancement." The term "organic" has the stem organ, which is a stem for 9 different words. The accumulated number of senses for "organ" is 39. With the same approach, we obtain 7 senses for 1 "soil" and 7 senses for "enhanc." Therefore, multiplication 39, 7, and 7 gives us the sense product value 1911.

4.2 Word Product

Word product is an alternative measure of query ambiguity after stemming. To compute the word product, we multiply the number of words mapped to each stem of a given query, which is formulated as:

$$\textit{word\_product} = \prod_{i=1}^{n} W_i \qquad (2)$$

W_i denotes the number of words mapped to a stem i, and n is the number of stems in a query. We assume that the query ambiguity increases as the word product increases. Consider the query "organic soil enhancement" in Figure 5. We find 9 words mapped to the stem "organ"; 3 words mapped to the stem "soil"; 5 words mapped to the stem "enhancement". Therefore the word product for the query is 105, the product of 9, 3, and 5.

4.3 Deplural-Stem Ratio

Deplural-stem ratio is a variation of sense product. It takes the ratio of the sense product of a depluralized query to the stemmed query. As the deplural-stem ratio increases, the query ambiguity after stemming increases. In the example illustrated in Figure 5, the deplural-stem ratio is the sense product of the depluralized query "organic soil enhancement" divided by the sense product of the stemmed query "organ soil enhanc". The deplural-stem ratio is computed as:

$$\textit{deplural-stem\_ratio} = \frac{\prod_{i=1}^{n} \sum_{j=1}^{m} Sm_j}{\prod_{i=1}^{n} \sum_{j=1}^{m} D_j} \qquad (3)$$

4.4 Idf-sum

The idf-sum is the sum of the inverse document frequency (IDF) of each word in the query. The IDF of a given word measures the importance of the word in the document collection. Queries with high values of IDF are more likely to return relevant documents from the collection. For example, the term "ZX-Turbo," describing a series of racing cars, has a high IDF and occurs only once in the entire TREC 2004 Robust Track collection. Therefore, searching the collection with the term "ZX-Turb" will return the only relevant document in the collection and achieve high precision and recall. The idf-sum is computed as:

$$\textit{idf\_sum} = \sum_{i=1}^{n} IDF_i \qquad (4)$$

IDF_i denotes the idf of each query term i and n is the number of words in a query. We assume that the query ambiguity decreases as the idf-sum increases. For the query "organic soil enhancement", the IDF for each term is 5.97082 (organic), 5.18994 (soil), and 4.86996 (enhancement). The idf-sum of the query is 16.0307.

4.5 Query Length

The length of the query is the number of words in a query.

4.6 Feature Validation

We performed simple linear regression on each feature as the first step to exclude ineffectual features. Table 2 demonstrates example results of simple linear regression from the MLT query model, using the MAP of stemmed queries as the dependent variable. We take the logarithm of the sense product and word product and the square root of the deplural-stem ratio (ds_ratio) to mitigate skewness of the data. We included all five ambiguity properties to construct a query-based selection model as they demonstrate statistical significance in prediction.

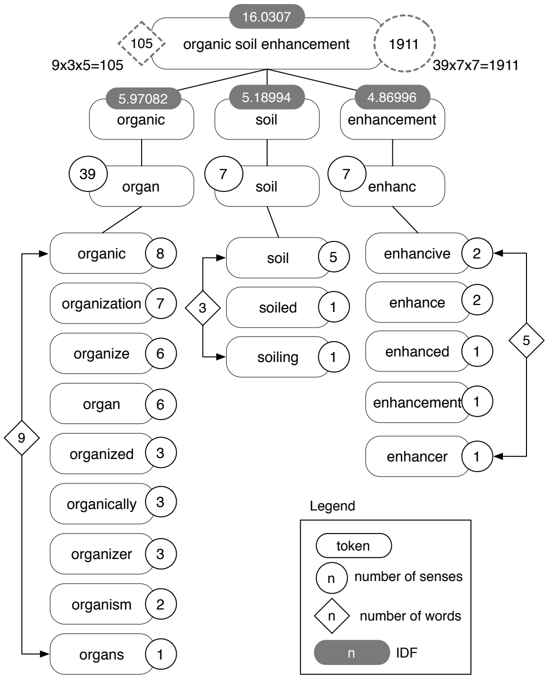

Figure 5: Example of ambiguity indicators on the query "organic soil enhancement"

Figure 6 demonstrates the distribution of ambiguity properties against the actual best text normalization technique. It is noted that stemming is the actual best method when a query has lower sense product, or lower word product, or a higher idf-sum. It implies that stemming is less likely to be the actual best method as a query is ambiguous. The results demonstrate the potential of utilizing ambiguity measures to select the actual best text normalization technique.

5 Query-based Selection Model

The cluster analysis in Section 3 suggests that the choice of text normalization technique should be made individually on each topic. The retrieval performance would be enhanced as we choose an appropriate text normalization technique for a given topic. Given the five ambiguity properties described in Section 4, we constructed Support Vector Regression (SVR) (Smola and Schlkopf, 2004) models to choose between stemming, depluralization, and not doing any text normalization for different queries. Regression models aim to discover the relationship between two random variables x and y. In our work, independent variable x is a vector of the five properties described in section 4: x = *(sense_product, word_product, deplural-stem_ratio, Idf-sum, length)*, and dependent variable y is the MAP score of a given topic. SVR has been successfully applied for many time series and function estimation problems. We utilized training data to construct multiple SVR models for each of nine combinations of query models (AND, OR, and MLT) and text normalization techniques (raw, depluralized, and stemmed queries). For example, the regression model for an MLT query model using stemmed queries is:

$$\begin{aligned} Map\_MLT\_stem = {} & 0.0853 \\ & - 0.0849 * length \\ & + 0.6286 * sense\_prod \\ & + 0.0171 * word\_prod \\ & - 0.0774 * gap\_ds \\ & + 0.4189 * idf\_sum \end{aligned}$$

For a given query model, MLT, for example, we utilized training data to construct three SVR models each to predict the MAP scores of raw queries, depluralized queries, and stemmed queries in the test set. We then compared the predicted MAP score of a query and selected the text normalization technique with the highest predicted score. Figure 5 illustrates our experiment framework on the query-based selection model. We used five-fold cross-validation to evaluate the performance of the selection model. For each iteration (fold) we used the 4 out of the 5 partitions as training data, constructing SVR models and using the remaining fifth partition for testing. We accumulated all testing results and computed one overall MAP score for evaluation. Table 3 shows the results of the five-fold cross-validation performed on 249 query topics provided by the TREC 2004 Robust Track. We utilized a paired t-test to determine the performance difference between the query-based selection model (hybrid model) and other three text normalization techniques. The results in Table 3 shows that the query-based selection model attained the highest MAP score and achieved significant improvement.

6 Conclusion and Future work

This paper evaluates the performance of stemming and depluralization on the TREC 2004 Robust track collection. We assume that the depluralization, as

Feature	Coefficient	R-square	P-value
length	-0.06811	0.07864	5.768e-05*
log(sense_prod)	-0.034692	0.1482	1.819e-08*
log(word_prod)	-0.04557	0.09426	9.78e-06*
sqrt(ds_ratio)	-0.021657	0.03738	0.006088*
idf_sum	0.008498	0.04165	0.003747*

Table 2: Results of simple linear regression on the MAP of stemmed queries in MLT query model.

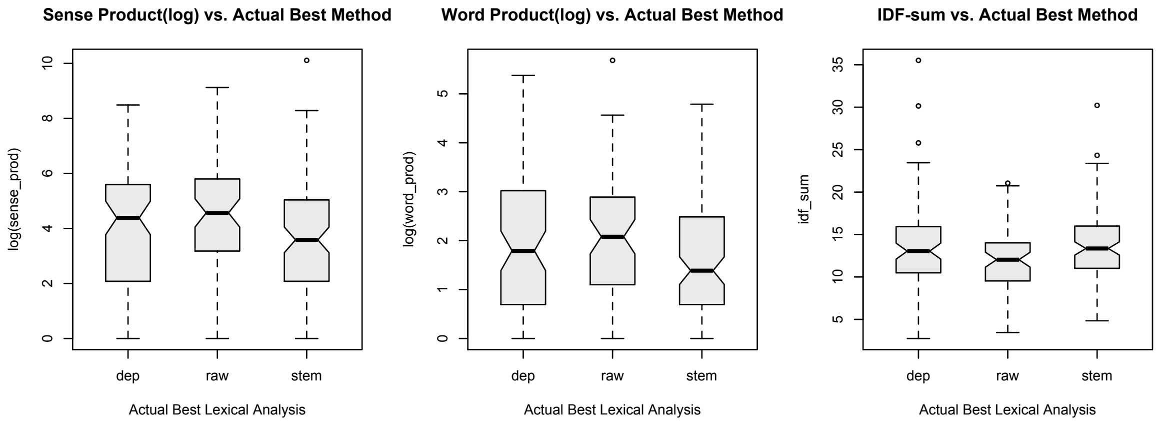

Figure 6: Boxplots of the distribution of three ambiguity properties in each actual best text normalization technique

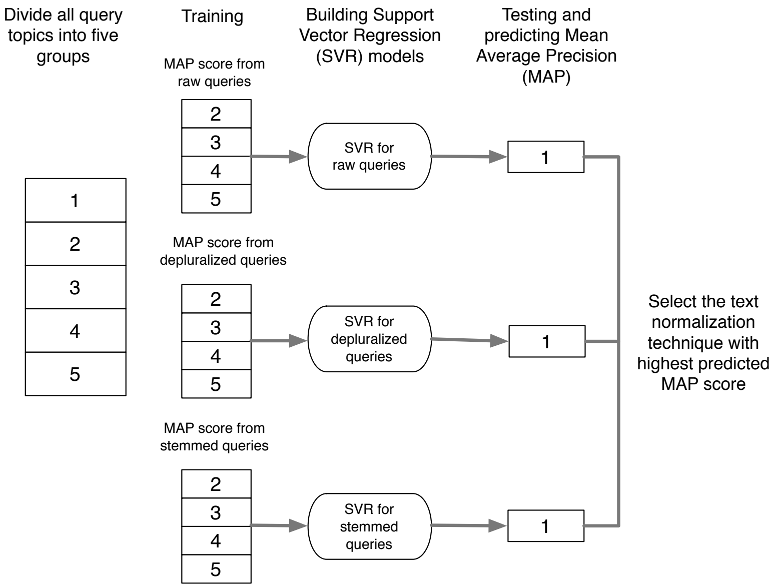
Divide all query topics into five groups
Training
Building Support Vector Regression (SVR) models
Testing and predicting Mean Average Precision (MAP)
MAP score from raw queries
2
3
4
5
SVR for raw queries
1
1
2
3
4
5
MAP score from depluralized queries
2
3
4
5
SVR for depluralized queries
1
Select the text normalization technique with highest predicted MAP score
MAP score from stemmed queries
2
3
4
5
SVR for stemmed queries
1

Figure 7: Five-fold cross validation on query-based selection model (hybrid model)

		Raw	**Dep**	**Stem**	**Hybrid**
AND	MAP	**0.1213**	**0.1324**	**0.1550**	**0.2094**
	p-value	<2.2e-16*	<2.2e-16*	<2.2e-16*	
OR	MAP	**0.1851**	**0.1922**	0.2069	**0.2131**
	p-value	1.286e-05*	0.0003815*	0.09	
MLT	MAP	**0.1893**	**0.1959**	0.2093	**0.2132**
	p-value	3.979e-05*	0.000939*	0.09677	

Table 3: Paired T-test was performed to examine the differences of each text normalization techniques (raw, depluralizer, and stemmer) and query-based selection model (hybrid model). Significant differences between models are labeled with *.

a low-level text-normalization technique, introduces less ambiguity than stemming and preserves more precise semantics of text. The experimental results demonstrate variable patterns, indicating that some topics performed better as a depluralized query than as a stemmed query. From Figure 4 in Section 3, we conclude that the choice of text normalization technique should be made individually on each topic. An effective query-based selection model would enhance information retrieval performance. The query-based selection model utilizes Support Vector Regression (SVR) models to predict the mean average precision (MAP) of each query from the ambiguity measures, and to choose an appropriate normalization technique based on these predictions. The selection is lightweight, requiring only analysis of the topic title itself against information readily available regarding the corpus (e.g, term idf values). We extracted 5 measures to quantify the ambiguity of a query: 1) sense product; 2) word product; 3) deplural-stem ratio; 4) idf-sum; 5) length of a query. The constructed query-based selection model demonstrate positive results on enhanced performance. The experiments reported here show that, even when the improvement is modest (1%), the selection model competes well with traditional approaches. To improve the model, future work may first explore and introduce more powerful features to the models, considering properties such as part of speech of text. Second, future work may explore the effect of noise and outliers in the data to improve the accuracy of the model. Finally, additional data mining techniques may be adopted in future work to further improve the prediction.

References

M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I.H. Witten. 2009. The WEKA Data Mining Software: An Update. *SIGKDD Explorations*, 1:10–18.

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

L. Kaufman and P.J. Rousseeuw. 1990. *Finding groups in data. An introduction to cluster analysis*. Wiley, New York.

Robert Krovetz. 2000. Viewing morphology as an inference process. *Artificial Intelligence*, 118(1-2):277–294, April.

Christopher D Manning and Hinrich Schtze. 1999. *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, Mass.

Paul McNamee, Charles Nicholas, and James Mayfield. 2008. Don't have a stemmer?: be un+concern+ed. In *Proceedings of the 31st International ACM SIGIR Conference on Research and Development in Information retrieval*, pages 813–814, Singapore, Singapore. ACM.

Ellen Riloff. 1995. Little words can make a big difference for text classification. In *Proceedings of the 18th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 130–136, Seattle, Washington, United States. ACM.

Alex J. Smola and Bernhard Schlkopf. 2004. A tutorial on support vector regression. *Statistics and Computing*, 14(3):199–222.

E. M. Voorhees. 2006. The TREC 2005 robust track. In *ACM SIGIR Forum*, volume 40, pages 41–48. ACM New York, NY, USA.