Estimating Query Difficulty for News Prediction Retrieval*

Nattiya Kanhabua
L3S Research Center
Leibniz Universität Hannover
Hannover, Germany
kanhabua@L3S.de

Kjetil Nørvåg
Dept. of Computer Science
Norwegian University of Science and Technology
Trondheim, Norway
noervaag@idi.ntnu.no

ABSTRACT

News prediction retrieval has recently emerged as the task of retrieving *predictions* related to a given news story (or a query). Predictions are defined as sentences containing time references to future events. Such future-related information is crucially important for understanding the temporal development of news stories, as well as strategies planning and risk management. The aforementioned work has been shown to retrieve a significant number of relevant predictions. However, only a certain news topics achieve good retrieval effectiveness. In this paper, we study how to determine the difficulty in retrieving predictions for a given news story. More precisely, we address the *query difficulty estimation* problem for news prediction retrieval. We propose different entity-based predictors used for classifying queries into two classes, namely, *Easy* and *Difficult*. Our prediction model is based on a machine learning approach. Through experiments on real-world data, we show that our proposed approach can predict query difficulty with high accuracy.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Retrieval models*; H.3.4 [**Information Storage and Retrieval**]: Systems and Software—*Performance evaluation (efficiency and effectiveness)*

General Terms

Algorithms, Experimentation, Performance

Keywords

Query difficulty estimation, Relevance ranking, News predictions, Future events

1. INTRODUCTION

What will happen in the eurozone after the financial crisis? How will health care change in the post-genomic society? When can renewable energy replace fossil fuels? These questions commonly arise when reading news stories, which reflect our anticipation and curiosity about the future. While future-related information helps people understand the temporal development of news stories, it can also be used for strategies planning to avoid/minimize disruptions, risks, and threats, or to maximize new opportunities. Canton [3] describes the describes the future trends that can influence our lives including an energy crisis, the global financial crisis, politics, health care, science, securities, globalization, climate changes, and technologies. Knowing about the future related to such topics is not only demanded by *individuals*, but also *organizations*, e.g., business firms or official governments.

In this paper, we address the retrieval and ranking task defined in [11], so-called *ranking related news predictions*. The objective of the task is to retrieve and rank *predictions* related to a given news story (or a query). Predictions are defined as sentences mentioning future dates, for instance, *Under the new rules, **eurozone** countries have to slash their budget deficits to a ceiling of 3% of GDP by **next year**,* or ***Cisco** reported it expects mobile web video traffic to increase 250-fold between 2011 and **2015**.* In the same study [11], they showed that nearly one third of news articles contain references to the future, as captured by time mentions of future dates in the articles.

While the approach proposed in [11] has been shown to retrieve a significant number of relevant predictions, the quality of result predictions vary greatly for different topics. In other words, only a certain type of queries achieves good retrieval effectiveness. Thus, we seek to improve the retrieval effectiveness for the worst performing queries, namely, entity queries, by studying *query difficulty estimation*. More precisely, we will focus on how to predict the quality of result predictions for a given topic or a news story, or estimating query difficulty for news prediction retrieval – to the best of our knowledge the first approach tackling this objective.

The main contributions in this paper are: 1) the first study of estimating query difficulty for news prediction retrieval, 2) proposing different predictors used for estimating query difficulty, and 3) extensive experiments for evaluating the proposed predictors using the New York Times Annotated Corpus in combination with queries selected from real-world future trends [3] and relevance assessments from [11].

The organization of the rest of the paper is as follows. In Section 2, we give an overview of related work. In Section 3, we describe the task of ranking related news prediction and the models for annotated documents, predictions, and queries. Then, we explain the problem of query difficulty estimation. In Section 4, we present a model for ranking result predictions. In Section 5, we propose novel predictors used for estimating query difficulty. In Section 6, we evaluate the proposed predictors and discuss results in detail. Finally, in Section 7, we conclude the paper.

*This is a corrected version of the printed paper.

CIKM'12, October 29–November 2, 2012, Maui, HI, USA.

2. RELATED WORK

The problem of query difficulty estimation [4] (also known as query performance prediction) has recently gained increasing interest from the IR community. Existing approaches to predicting query performance can be categorized wrt. two aspects [7]: 1) time of predicting (pre/post-retrieval) and 2) objective of task (difficulty, query rank, effectiveness). Pre-retrieval predictors work independently from a specific retrieval model and result documents, and such predictors are preferred to post-retrieval based methods because they are based solely on query terms, collection statistics and possibly external knowledge, such as WordNet or Wikipedia.

By measuring the *specificity* of query terms, the effectiveness of a query can be estimated by assuming that the more specific a query, the better effectiveness it will achieve. In order to determine the specificity, different heuristic-based predictors have been proposed, for example, the averaged length of a query [12], the averaged inverse document frequency [5] and the averaged inverse collection term frequency [8]. The summed collection query similarity [15] employs both term frequencies and inverse document frequencies. Another approach for estimating query difficulty is to measure *query ambiguity*. An example of an *ambiguity* based predictor is a set coherence score [9] measuring the ambiguity of a query by calculating the similarity between all documents that contain the query term.

The predictors presented above ignore *term relatedness* among query terms. To measure the relationship between two terms, pointwise mutual information (PMI) is computed as suggested in [7]. PMI measures the term relationship by observing co-occurrence statistics of terms in a document collection. Two PMI-based predictors are proposed in [7] including the averaged PMI value and the maximum PMI value of all query term pairs. More detailed descriptions of different approaches to query difficulty estimation can be found in the book by Carmel and Yom-Tov [4], and references therein.

The problem of future information retrieval was first presented by Baeza-Yates [1]. He proposed to extract temporal mentions of future events from news articles, index and retrieve such information using a probabilistic model. A document score was computed by multiplying a *keyword* similarity and a time confidence, i.e., a probability that the future events will actually happen. Jatowt et al. [10] proposed an analytical tool for extracting, summarizing and aggregating future-related events from news archives using a clustering method. In recent work, Kanhabua et al. [11] proposed the novel task of ranking related news predictions with the main goal of improving the retrieval effectiveness of future information (as captured by mentions of future dates in news articles). They proposed a ranking model based on a learning-to-rank technique, which is learned using different features. None of aforementioned work addresses the problem of *query difficulty estimation* for future information retrieval, which is the topic of this paper.

3. PRELIMINARIES

In this section, we briefly describe the task of *ranking related news predictions*. Then, we outline the models for annotated documents, predictions, and queries. Finally, we describe how to perform query difficulty estimation.

3.1 Ranking Related News Predictions

The task of ranking related news predictions was first proposed in [11]. Predictions can be automatically extracted from a *temporal document collection*, (e.g., news archives, company websites, financial reports, or blogs) using a series of annotation processes including tokenization, sentence extraction, part-of-speech tagging, named entity recognition, and temporal expression extraction. The results are predictions or sentences annotated with named entities and future dates.

Instead of having a user's information need explicitly provided, a query will be automatically generated from the *news article being read* by the user. For example, a query can be top-m entities or top-n terms extracted from the news article. For a given news article, predictions will be retrieved and ranked by the degree of relevance. As defined in [11], a prediction is "relevant" if it is future information about the topics of the news article. Note that, there is no specific instructions about how the dates involved are related to relevance. However, predictions extracted from more recent documents are assumed to be more relevant.

3.2 Annotated Document Model

The document collection used in this work is a collection of news articles defined as $C = \{d_1, \ldots, d_n\}$. A news article is represented as a bag-of-words, $d = \{w_1, \ldots, w_n\}$. The publication time of d is denoted by the function *time*(d). Each document d is associated to an annotated document $\hat{d}$ composed of three parts: $\hat{d}_e$ is a set of named entities $\hat{d}_e = \{e_1, \ldots, e_n\}$, where each entity e_i is a type of person, location, or organization; $\hat{d}_t$ is a set of annotated temporal expressions $\hat{d}_t = \{t_1, \ldots, t_m\}$ and $\hat{d}_s$ is a set of sentences $\hat{d}_s = \{s_1, \ldots, s_z\}$. Later in the paper, we will propose predictors used for estimating query difficulty, which are based on these annotated documents.

3.3 Prediction Model

A prediction p is associated with its *parent document* d^p, where p is extracted from, and each prediction p is represented as a sentence with multiple fields/values including: a prediction's unique number (ID), the unique number of d^p (PARENT_ID), the title of d^p (TITLE), annotated entities p_{entity} in p (ENTITY), future dates p_{future} in p (FUTURE_DATE), the publication time of d^p (PUB_DATE), the sentence text of p (TEXT), and surrounding sentences of p (CONTEXT).

3.4 Query Model

A query q is extracted automatically from a news article d^q being read, where q is composed of two parts: keywords q_{text}, and the time of query q_{time}. The keywords q_{text} can be generated from d^q in three ways, resulting in three types of queries: 1) entity query (a list of *top-m* entities ranked by frequency), 2) term query (*top-n* terms ranked by term weighting, i.e., TF-IDF), and 3) combined query (combining both *top-m* entities and *top-n* terms).

In this work, we are interested in *entity queries* only, where representing a query using *top-m* entities performed worst among other query types as shown in [11]. Thus, we seek to improve the retrieval effectiveness for entity queries by performing *query difficulty estimation* during the retrieval stage so that particular actions can be taken to improve the overall performance. Consider a news article d^q about "President Bush and the Iraq war", the keyword part of an entity query q can be represented as $q_{text} = \langle \textit{George Bush}, \textit{Iraq}, \textit{America} \rangle$. During retrieval, q_{text} will be matched with the ENTITY field of the predictions.

The time q_{time} are two *time constraints* used for retrieving predictions. First, only predictions that are *future* relative to the publication time of query's parent article, or *time*(d^q) will be retrieved. Second, those predictions must belong to news articles published before *time*(d^q). Both time constraints are represented using a time interval, i.e., $[t_b, t_e]$, where t_b is a beginning time point, t_e is an ending time point, and t_e is greater than t_b. In all cases, the first constraint is $(\textit{time}(d^q), t_{max}]$, and the second constraint is $[t_{min}, \textit{time}(d^q))]$, where $(\textit{time}(d^q), t_{max}] = [\textit{time}(d^q), t_{max}] - \{\textit{time}(d^q)\}$,

and t_{max} and t_{min} are the maximum time in the future and the minimum time in the past respectively. During retrieval, predictions will be retrieved by matching the first constraint with the field FUTURE_DATE and the second constraint with the field PUB_DATE.

3.5 Query Difficulty Estimation

The task of query difficulty estimation can be viewed as a classification problem. Queries will be labeled into *predefined classes* based on how well a particular ranking model performs. In this work, we consider two classes of queries: *Easy* and *Difficult*. The query difficulty can be determined using the retrieval effectiveness, such as, the Mean Average Precision (MAP). A query achieves the *higher* MAP wrt. a particular ranking model is considered the *easier* query. On the contrary, the *lower* MAP a query achieves, the *more difficult* the query is.

The prediction quality is highly dependent on a retrieval model because the effectiveness is dependent on a specific retrieval approach. In addition, the prediction quality also depends on a dataset and a document collection used for retrieval [4]. Thus, we take into account *prediction robustness* by employing several ranking models (cf. Section 4) in determining the difficulty of a given query or topic. Those models can also be regarded as different runs.

We follow a similar approach for identifying classes of queries as presented in [13]. In order to label queries as *Easy* or *Difficult*, we use a condition for splitting queries into two groups. For a given query q, we measure MAP wrt. all ranking models and determine whether the average of MAP (denoted *avgMAP*) and the standard deviation of MAP (denoted *stdMAP*) exceed the respective thresholds ϵ and ω or not.

if *avgMAP*$(q) \geq \epsilon$ **and** *stdMAP*$(q) \geq \omega$ **then**
  $q_{class} = Easy$
else
  if *avgMAP*$(q) < \epsilon$ **and** *stdMAP*$(q) < \omega$ **then**
    $q_{class} = Difficult$
  end if
end if

4. RANKING MODEL

In this section, we will present features and two models used for ranking result predictions, which are based on a feature-based ranking model. The features can be categorized into two classes: 1) pre-retrieval and 2) post-retrieval, where both classes are obtained from entity information at different retrieval stages. Note that, the features to be presented are commonly employed in an entity ranking task [2, 6] but used in other context. In this work, we use entity-based features in order to capture the semantic similarity between q and p.

Pre-retrieval features are extracted from annotation data of *a query article* (a news article being read d^q) and thus independent from retrieval and the ranked list of result predictions. The features in this class include *senPos, senLen, cntSenSubj, cntEvent, cntFuture, cntEventSubj, cntFutureSubj, timeDistEvent, timeDistFuture* and *tagSim*. The first feature *senPos* gives the position of the 1^{st} sentence where e occurs in d^p. *senLen* gives the length of the first sentence of d that contains e. *cntSenSubj* is the number of sentences where e is a subject. *cntEvent* is the number of event sentences (or sentences annotated with dates) of e.

cntFuture is the number of sentences with a mention of a future date. *cntEventSubj* is the number of event sentences where e is a subject. *timeDistEvent* is a measure of the distance between e and all dates in d^p. *timeDistFuture*(e, d^p) is the distance of e and all future dates in d^p computed similarly to *timeDistEvent*. *tagSim* is the string similarity between e and an entity tagged in d^p. *tagSim* is only applicable for a collection provided with manually assigned tags (e.g., the New York Times Annotated Corpus).

Post-retrieval features are, in contrast to the previous class, extracted from the annotation data of *result predictions* including *isSubj* and *timeDist*. *isSubj*(e, p) is 1 if e is a subject with respect to a prediction p, and *timeDist*(e, p) is a distance of e and all future dates in p computed similarly to *timeDistEvent*.

A set of all features $\mathcal{F}$ presented previously are parameter-free, and their values will be normalized to range from 0 to 1. The detailed computation of different features can be found in [11].

We propose two different ranking models that linearly combine two normalized scores:

$$S'(q,p) = (1-\lambda) \cdot S_{term}(q,p) + \lambda \cdot S_{single}(q,p) \quad (1)$$

$$S''(q,p) = (1-\lambda) \cdot S_{term}(q,p) + \lambda \cdot S_{combined}(q,p) \quad (2)$$

where the mixture parameter λ indicates the importance of term-based similarity and entity-based similarity. The term-based similarity $S_{term}(q,p)$ can be measured using any of existing text-based weighting functions, e.g., TF-IDF or a unigram language model. The entity-based similarity can be computed using the features presented above as a single feature $S_{single}(q,p)$, or a combination of multiple features $S_{combined}(q,p)$. All similarity scores will be normalized, e.g., divided by the maximum scores, before generating the final scores: $S'(q,p)$ and $S''(q,p)$.

The score for multiple features are computed by linearly combining the scores of three different features as follows.

$$S_{combined}(q,p) = (1-\alpha-\beta) \cdot f_i + \alpha \cdot f_j + \beta \cdot f_k \quad (3)$$

where each individual feature is a member of a set of all features: $\{f_i, f_j, f_k\} \subset \mathcal{F}$. α and β are mixture parameters giving a weight to the score of each feature, where $\alpha + \beta < 1$.

5. QUERY DIFFICULTY PREDICTORS

In this section, we present our methodology for estimating query difficulty, which is based on a machine learning approach. We will learn a classification model using features, so-called *predictors*, in order to classify queries into two classes, i.e., *Easy* and *Difficult*.

We propose 10 post-retrieval predictors that are derived from analyzing top-k retrieved predictions: *cntEntity, avgEntityPerPredict, distinctEntity, avgPredictPerEntity, cntPeople, percentPeople, cntOrg, percentOrg, cntLoc*, and *percentLoc*. Note that, these features used for predicting query difficult are *different* from those used for ranking result predictions (presented in Section 4).

Our proposed predictors are aimed at capturing the *ambiguity* of a query by analyzing entities (i.e., people, organization, and location) in the top-k retrieved predictions. To the best of our knowledge, the proposed predictors have never been employed in a similar task before. The description of the proposed predictors is shown in Table 1. The actual values of all predictors will be calculated with respect to top-k retrieved predictions, where the k value will be varied in the experiments

6. EXPERIMENTS

The New York Times Annotated Corpus (with 1.8 million news articles from 1987 to 2007) was used as a temporal document collection. Documents were annotated and predictions were extracted using different NLP tools as follows. We extracted sentences and

Table 1: Description of the post-retrieval predictors.

Predictor	Description
cntEntity	the number of all entities
avgEntityPerPredict	an average of entities per prediction
distinctEntity	the number of distinct entities
avgPredictPerEntity	an average of predictions per distinct entity
cntPeople	the number of *people* entities
percentPeople	the percentage of *people* in all entities
cntOrg	the number of *organization* entities
percentOrg	the percentage of *organization* in all entities
cntLoc	the number of *location* entities
percentLoc	the percentage of *location* in all entities

performed part-of-speech tagging using OpenNLP. The SuperSense tagger was used for named entity recognition and the TARSQI Toolkit was used for extracting temporal expressions. The Apache Lucene search engine was employed for both indexing and retrieving predictions.

We used future-related queries and relevance assessments from the previous work [11]. The dataset is composed of 42 query news articles related to future topics and 4,888 manually evaluated pairs of query/prediction. In this work, the actual queries or "entity queries" used for retrieving predictions were extracted from these query news articles.

Parameters used in the experiments were set as follows. We represented an entity query using the number of entities m = 11 as recommended in [11]. For the ranking models, we generated all possible runs by varying the values for λ, α, and β from 0 to 1 by increment of 0.1. For a given query, we measured the retrieval effectiveness using the Mean Average Precision (MAP).

In order to label a query into two classes (*Easy* and *Difficult*), we determined whether *avgMAP* and *stdMAP* are greater than thresholds ϵ and ω or not. In this work, we used $\epsilon = 0.4$ and $\omega = 0.03$, as we observe empirically. The Weka implementation [14] was used for modeling the query difficulty prediction as a classifier, which was learned using several algorithms: decision tree, Naïve Bayes, neural network and SVM, using 10-fold cross-validation with 10 repetitions. We measured statistical significance using a t-test with $p < 0.05$. In the tables, bold face indicates statistically significant difference from the respective baseline.

Classification results. The baseline method for query classification is the majority classifier. The accuracy of the baseline is 0.79. Table 2 shows the accuracy of the best-performing classification algorithm on each predictor. The combination of all predictors is denoted *ALL*. We varied the number of top-k retrieved documents in order to study how a k-value affect the classification performance, namely, $k = 10, 25, 50, 75$ and 100.

The results show that the predictor *avgEntityPerPredict* performs best almost in every k's value when comparing with other predictors. The other predictors do not achieve better accuracy among them. The combination of all features gives the best result with the accuracy of 0.92. Hence, the combined predictor can be used for estimating query difficulty with high accuracy. We do not observe any trend in performance for different k's values.

Table 2: Accuracy of query classification.

Predictor	Top-k				
	10	*25*	*50*	*75*	*100*
cntEntity	.76	.76	.76	.76	.78
avgEntityPerPredict	**.83**	.78	**.83**	**.83**	**.85**
distinctEntity	.75	.73	.74	.74	.75
avgPredictPerEntity	.76	.76	.80	.80	.81
cntPeople	.76	.76	.78	.78	.78
percentPeople	.77	.77	.77	.77	.77
cntOrg	.76	.81	.79	.81	.80
percentOrg	.76	.76	.79	.77	.77
cntLoc	.76	.76	.76	.76	.76
percentLoc	.78	.76	.81	.76	.76
ALL	**.92**	.83	.86	.79	.86

7. CONCLUSIONS

In this paper, we have proposed a machine learning approach to estimate query difficulty for the ranking related news prediction task. Our proposed predictors are based on entity information, which can be extracted from annotation data of news articles. Through experiments using real-world dataset, we showed that our proposed approach is able to predict two classes of query difficulty with high accuracy.

8. REFERENCES

[1] R. A. Baeza-Yates. Searching the future. In *Proceedings of SIGIR workshop on mathematical/formal methods in information retrieval MF/IR 2005*, 2005.

[2] K. Balog, L. Azzopardi, and M. de Rijke. A language modeling framework for expert finding. *Inf. Process. Manage.*, 45(1):1–19, 2009.

[3] J. Canton. *The Extreme Future: The Top Trends That Will Reshape the World in the Next 20 Years*. Plume, 2007.

[4] D. Carmel and E. Yom-Tov. *Estimating the Query Difficulty for Information Retrieval*. Morgan & Claypool Publishers, 2010.

[5] S. Cronen-Townsend, Y. Zhou, and W. B. Croft. Predicting query performance. In *Proceedings of SIGIR'2002*, 2002.

[6] G. Demartini, A. P. de Vries, T. Iofciu, and J. Zhu. *Overview of the INEX 2008 Entity Ranking Track*. 2009.

[7] C. Hauff, L. Azzopardi, and D. Hiemstra. The combination and evaluation of query performance prediction methods. In *Proceedings of ECIR'2009*, 2009.

[8] B. He and I. Ounis. Inferring query performance using pre-retrieval predictors. In *Proceedings of SPIRE'2004*, 2004.

[9] J. He, M. Larson, and M. de Rijke. Using coherence-based measures to predict query difficulty. In *Proceedings of ECIR'2008*, 2008.

[10] A. Jatowt, K. Kanazawa, S. Oyama, and K. Tanaka. Supporting analysis of future-related information in news archives and the web. In *Proceedings of JCDL'2009*, 2009.

[11] N. Kanhabua, R. Blanco, and M. Matthews. Ranking related news predictions. In *Proceeding of SIGIR'2011*, 2011.

[12] J. Mothe and L. Tanguy. Linguistic features to predict query difficulty - a case study on previous trec campaigns. In *Proceedings of SIGIR Workshop on Predicting Query Difficulty - Methods and Applications*, SIGIR'2005, 2005.

[13] A.-M. Vercoustre, J. Pehcevski, and V. Naumovski. Topic difficulty prediction in entity ranking. In *Proceedings of INEX'2009*, 2009.

[14] I. H. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques, Second Edition*. Morgan Kaufmann, 2005.

[15] Y. Zhao, F. Scholer, and Y. Tsegay. Effective pre-retrieval query performance prediction using similarity and variability evidence. In *Proceedings of ECIR'2008*, 2008.