Published in Multilingual Information Access in South Asian Languages, 23-37, 2013
which should be used for any reference to this work

Ad Hoc Retrieval with Marathi Language

Mitra Akasereh and Jacques Savoy

Computer Science Department,
University of Neuchatel,
Rue Emile Argand 11, 2000 Neuchatel, Switzerland
{Mitra.Akasereh,Jacques.Savoy}@unine.ch

Abstract. Our goal in participating in FIRE 2011 evaluation campaign is to analyse and evaluate the retrieval effectiveness of our implemented retrieval system when using Marathi language. We have developed a light and an aggressive stemmer for this language as well as a stopword list. In our experiment seven different IR models (language model, DFR-PL2, DFR-PB2, DFR-GL2, DFR-I(n_e)C2, *tf idf* and Okapi) were used to evaluate the influence of these stemmers as well as *n*-grams and trunc-*n* language-independent indexing strategies, on retrieval performance. We also applied a pseudo relevance-feedback or blind-query expansion approach to estimate the impact of this approach on enhancing the retrieval effectiveness. Our results show that for Marathi language DFR-I(n_e)C2, DFR-PL2 and Okapi IR models result the best performance. For this language trunc-*n* indexing strategy gives the best retrieval effectiveness comparing to other stemming and indexing approaches. Also the adopted pseudo-relevance feedback approach tends to enhance the retrieval effectiveness.

Keywords: Marathi information retrieval, retrieval effectiveness with Indian languages, FIRE evaluation campaign, automatic indexing.

1 Introduction

One of our main objectives in the IR group of University of Neuchâtel is to design, implement and evaluate various indexing and search strategies that work with different non-English languages (monolingual IR). More specifically, in this part we begin with less frequently used languages (and new from an IR perspective), such as Persian, Turkish, Polish, Hindi, Marathi, Bengali and other Indian languages (e.g., Punjabi, Tamil, Telugu) [1]. This set of languages covers various branches of the Indo-European family, while we also tackle popular European [2] as well as Far-East (e.g., Chinese, Japanese, and Korean) [3] languages in order to provide a basis of comparison for our tests. Our objective also includes bilingual and multilingual IR systems. In our participation in the FIRE campaign (www.isical.ac.in/~fire/), our main motivation is to promote new tools and to evaluate and improve existing ones for monolingual IR when facing with Hindi, Marathi and Bengali languages. We applied our tests on the three above mentioned languages while in this paper we talk only about Marathi language. The reason is that our results for Bengali and Hindi

languages were not completely reliable due to some mistakes while applying stopword removal on these two languages.

The rest of this paper is organized as follows: Section 2 contains a brief introduction to Marathi language, Section 3 gives an overview of the corpus used in the FIRE-2011 *ad hoc* task. Section 4 represents an overview of our experiment setup and introduces different IR models used in the experiment, the developed stopword list as well as applied stemming and indexing strategies and finally the evaluation method used to evaluate our results. Section 5 presents the results obtained during the experiment and contains the analysis on the usage of different query formulations, different stemming and indexing strategies, various IR models and the impact of query expansion on obtained results. Finally, Section 6 concludes the experiment.

2 Marathi Language

Within the Indian languages studied in FIRE 2011 evaluation campaign, the Marathi owns a special place due to its complex inflectional morphology compared to the Hindi or Bengali languages. In fact, the Marathi grammar has three genders (masculine, feminine, and neuter), two numbers (singular, and plural) and eight grammatical cases (nominative, accusative, genitive, dative, ablative, locative, instrumental, and oblique). At the limit, we may have $3 \times 2 \times 8 = 48$ different suffixes. As for other Indo-European languages, however, this theoretical limit is not reached and several combinations of gender, number and grammatical case own the same suffix. Nevertheless, this highly inflected language raises challenges when designing a light stemmer. To be precise, this is not directly related to the number of possible suffixes. For example, the Hungarian morphology owns 23 grammatical cases while the Finnish language has 15. For an IR point of view, developing an effective Hungarian stemmer is possible [4] while for the Finnish language none algorithmic stemmer is able to produce useful stems. Within this language, adding a given suffix may alter letters inside the word to facilitate its pronunciation (and governed by various phonetic rules such as the vowel harmony) [5]. The linguistic equation "stem + suffix = surface word" possesses numerous exceptions in Finnish while it is relatively direct in the Hungarian language.

As for other languages, the Marathi morphology owns some irregularities (e.g., in plural form, the noun "child" gives "children" and not "childs"). In the best of our knowledge, these exceptions are less frequent than in the Finnish language but more frequent than in other languages (e.g., Hungarian, or German).

3 Overview of the Corpus

The test-collection used for this experiment is the collection made available during the FIRE 2011. The Marathi collection is a collection of about 618 MB of data made up of 99,270 news articles with the average of 266 terms per document (215 terms per document after stopword removal). The articles are extracted from two sources: Esakal between April 2007 to september 2007 and Maharashtratimes between

September 1st 2004 and December 31st 2006. The corpus is coded in UTF-8 and each article is marked up using following tags:

<DOC> : Starting tag of a document.
<DOCNO> </DOCNO> : Contains document identifier.
<TEXT> </TEXT> : Contains document text.
</DOC> : Ending tag of a document.

In this corpus we can find 50 topics (from Topic #126 to Topic #175). Among which fourteen topics have no relevant document in the collection (Topic #126, #129, #132, #137, #141, #145, #151, #154, #155, #156, #158, #159, #160 and #162). The rest thirty six topics have a total number of 354 relevant documents with mean of 9.83 items/topic and a median of 4 (standard deviation 14.26). Topics #130, #133, #139, #143, #146, #149, #150, #164, and #167 own one relevant item which is the smallest number of pertinent documents. Topic #170 with 62 relevant items has the greatest number of relevant documents.

Following the TREC model [6], each topic is divided into three sections: the title (T) which is a brief title, the description (D) that gives a one-sentence description, and the narrative part (N) which specifies the relevance assessment criteria. Topic #145 is shown below as an example. This topic holds "Benazir Bhutto murder" in its title, "Benazir Bhutto murder enquiry" in its description, while more details on the subject is given in its narrative section.

<top lang='mr'>
<num>145</num>
<title>बेनजीर भुट्टो यांची हत्या</title>
<desc>बेनजीर भुट्टो यांच्या हत्येची चौकशी</desc>
<narr>प्रासंगिक लेखांत बेनजीर भुट्टो यांच्या हत्येची चौकशी आणि चौकशीच्या कार्यपद्धतीबद्दल (चौकशीच्या प्रणालीबद्दल) विविध लोक, गट आणि संघटना ह्यांचे विचार व मते यांविषयी माहिती असावी.</narr>
</top>

4 Experiment Architecture

This section describes the setup of our experiment. Section 4.1 describes the adopted IR models, Section 4.2 explains the stopword list used for stopword removal, Section 4.3 describes the applied stemming and indexing strategies while Section 4.4 explains the measurements used to evaluate our system.

4.1 IR Models

In the experiment we analysed and compared different stemming and indexing strategies. To achieve this, seven different IR models are implemented and evaluated. The models are the following:

The first model is the classical *tf idf* model, where the weight for each indexing term t_i is the product of its term frequency in the document (tf_{ij}) and the logarithm of its inverse document frequency ($idf_j = \log\left[\frac{n}{df_j}\right]$ where n indicates the number of documents in the collection and df_j the number of documents which indexed the term t_i). The index weights normalized using cosine normalization. To compute the similarity between a document and a given query we have adopted the inner product given in Equation 1.

$$sim(d_i, q) = \sum_{t_i \in q} w_{ij} \cdot w_{qj} \tag{1}$$

As the first probabilistic model we have adopted the Okapi model (BM25) [7]. To evaluate the score of the similarity between the query and the document the Okapi function is described in Equation 2. In this formulation l_i indicates the length of document d_i (number of indexing terms). In our experiment the *advl* (average document length) is set to 265, b to 0.55 and k_1 to 1.2.

$$sim(d_i, q) = \sum_{t_i \in q} qtf_j \cdot \log\left[\frac{n - df_j}{df_j}\right] \cdot \frac{(k_1 + 1) \cdot tf_{ij}}{K + tf_{ij}} \tag{2}$$

$$K = k_1 \cdot \left((1 - b) + b \cdot \frac{l_i}{advl}\right)$$

As other probabilistic models, we have used the DFR-PL2, DFR-I(n_e)C2, DFR-PB2 and DFR-GL2. These models are derived from the Divergence From Randomness (DFR) family [8]. The indexing term weight (weight of term t_j in document d_i) is calculated as:

$$w_{ij} = Inf_{ij}^1 \cdot Inf_{ij}^2 = -\log_2\left(Prob_{ij}^1(tf_{ij})\right) \cdot \left(1 - Prob_{ij}^2(tf_{ij})\right) \tag{3}$$

In this experiment the DFR-PL2 is implemented as in Equations 4 and 5.

$$Prob_{ij}^1 = \frac{e^{-\lambda_j} \cdot \lambda_j^{tfn_{ij}}}{tfn_{ij}!} \tag{4}$$

$$Prob_{ij}^2 = \frac{tfn_{ij}}{tfn_{ij} + 1} \tag{5}$$

where:

- tf_n is the normalized term frequency
- $\lambda_j = \frac{tc_j}{n}$ (tc_j is the number of occurrence of term t_i in the collection and n is the number of documents in the corpus)
- $tfn_{ij} = tf_{ij} \cdot \log_2\left(1 + \frac{c \cdot mean\_dl}{l_i}\right)$ (in this experiment c is set to 1.5 and $mean\_dl$ (average document length) to 265)

For the DFR-PB2 model, the $Prob_{ij}^1$ is calculated as mentioned in Equation 4 and $Prob_{ij}^2$ as follows:

$$Prob_{ij}^2 = 1 - \frac{tc_j + 1}{df_j(tfn_{ij} + 1)} \tag{6}$$

DFR- I(n_e)C2 is defined by the same $Prob_{ij}^2$ as in Equation 6 and:

$$Inf_{ij}^1 = tfn_{ij} \cdot \log\left[\frac{n+1}{n_e + 0.5}\right] \tag{7}$$

$$n_e = n \cdot \left(1 - \left(\frac{n-1}{n}\right)^{tc_j}\right)$$

In the DFR-GL2 model $Prob_{ij}^1$ is defined as follows while $Prob_{ij}^2$ is defined as in Equation 5.

$$Prob_{ij}^1 = 1 - \left[\frac{1}{1+\lambda_j}\right] \cdot \left[\frac{\lambda_j}{1+\lambda_j}\right]^{tfn_{ij}} \tag{8}$$

Finally we have also implemented one approach based on language model (LM) [9]. In this case, the underlying paradigm is based on a non-parametric probabilistic model. We have opted the model suggested by Hiemstra [10] using the Jelinek-Mercer smoothing [11] scheme as shown in Equation 9.

$$P(d_i|q) = P(d_i) \cdot \prod_{t_j \in q} \left[\lambda_j \cdot P(t_j|d_i) + (1 - \lambda_j) \cdot P(t_j|C)\right] \tag{9}$$

where:

- $P(t_j|d_i) = \frac{tf_{ij}}{l_i}$
- $P(t_j|C) = \frac{df_j}{lc}$ $(lc = \sum_k df_k)$
- λ_i is a smoothing factor (here set to 0.30 for all index terms)
- lc is an estimation of the corpus C length.

4.2 Stopword List

Different words do not have the same importance in describing the semantic content of a document (or a query). Therefore it is usually a good practice to remove very frequent terms having no precise meaning (stopwords). Accordingly before applying the indexing strategies we eliminate the stopwords. To do so we used a stopword list. This list is generated using the approach explained in [12] containing 99 terms. The list is freely available on http://www.unine.ch/info/clef/.

4.3 Stemming and Indexing Strategies

To represent the documents and the topics, different automatic indexing methods can be applied. As a first strategy, we can use some language-independent indexing representations. The two possible strategies are the *n*-gram and trunc-*n* approaches. *N*-grams approach is the act of producing, for each word, the overlapping sequences of *n* characters [13]. With the word "computer" and defining *n*=3, we obtain {com, omp, mpu, put, ute, ter} indexing terms. The trunc-*n* is the process of truncating a word by keeping its first *n* characters and cutting of the remaining letters. With our previous example we obtain "comp" with n=4. In our experiment different values for *n*, for both *n*-gram and trunc-*n* are tested to find which value of *n* gives the best performance.

Having some morphological variations do not usually change the meaning of a word. For example a document with the word "houses" may be a good answer to a query containing the word "house". As a way to conflate word variations under the same form, we can apply a stemmer that removes the final letters of a word. To do so, in this experiment a light suffix-stripping algorithm is used which removes the inflectional suffixes. In our implementation, the removal is focused on nouns and adjectives and verbal suffixes are not taken into account. The reason for ignoring the verbal suffixes is mainly due to the following hypothesis. We believe that in a given text verbs convey less important semantic information than nouns and adjectives thus the retrieval based on match between different verbal forms is less useful. There are more details on this assumption in our previous experiments on other languages [14]. On the other hand, considering the fact that our light stemmer does not take into account part of speech and it does not apply any morphological analysis, verbal suffix removal would not help the retrieval effectiveness and might even reduce the mean average precision. Moreover previous experiments show that stemmers based on deep morphological analysis do not give better retrieval results than simpler light stemmers [15], [16], [17]. As a variant, we will also apply a more aggressive stemmer which removes also some derivational suffixes apart from inflectional ones. In this case we can conflate words like "computational" and "computer" under the same root.

4.4 Evaluation

To evaluate the retrieval performance we have adopted the mean average precision (MAP) measurement (as calculated by TREC_EVAL program [18] based on the 1000 retrieved documents per query). Using the average means that we attach the same importance to all queries. It is important to mention that in our calculations the queries with no relevant items were not taken into account. So for Marathi language we considered only 36 queries (for which there is at least one relevant document in the collection) while the official measure takes the whole 50 queries into account. As a result there are some differences between values presented in this paper and the ones computed according to the official measure.

5 Results and Analysis

This section evaluates and analyses the results obtained during our experiment. Section 5.1 contains the overall results obtained during the experiment. Sections 5.2 and 5.3 discuss the performance of the seven adopted IR models and different query formulation (T, TD and TDN). Section 5.4 compares the different stemming strategies used in the experiment (light and aggressive) with no stemming approach. While Section 5.5 discusses the effectiveness of using indexing strategies (n-gram and trunc-n) and compare the results obtained for different values for n. Finally Section 5.6 shows the results when applying a blind-query expansion technique.

5.1 Experiment Results

The MAPs of our different experiments are depicted in Table 1 (Title only), Table 2 (Title and Description) and Table 3 (Title, Description and Narrative). In these tables, we can find the retrieval effectiveness values obtained by applying different stemming and indexing approaches to seven different IR models. For indexing approaches like n-gram and trunc-n different values of n are selected and evaluated in order to define which value for n gives a better overall performance.

The first row of the three tables shows the results when the retrieval is done without applying either a stemming strategy (NoStem) or stopword removal (NoSL). While the next row shows the retrieval performance where stopword removal is applied but still stemming is ignored. The next two rows show the effects of applying a light and a more aggressive stemmer. Afterwards the results of performing various n-gram approaches with two different values for n are depicted. Finally the last three rows report the results for trunc-n approach with three different values of n.

Table 1. MAP of different IR models and different stemmers for T query formulation

	Mean Average Precision (T)							
	Okapi	**DFR-I(n_e)C2**	**DFR-PL2**	**DFR-GL2**	**LM**	***tf idf***	**DFR-PB2**	**Average**
NoStem/NoSL	0.2038	0.2147	0.2028	0.2038	0.2035	0.1422	0.2074	0.1969
NoStem	0.2044	0.2128	0.2013	0.1957	0.2011	0.1418	0.2057	0.1947
Light Stem.	0.2044	0.2128	0.2013	0.1957	0.2011	0.1418	0.2057	0.1947
Aggressive	0.2044	0.2129	0.2015	0.1961	0.2011	0.1418	0.2057	0.1948
3-grams	0.2733	0.2635	0.2579	0.2574	0.2131	0.1700	0.1297	0.2236
4-grams	0.2454	0.2433	0.2424	0.2476	0.2379	0.1607	0.2249	0.2289
trunc-3	0.2239	0.2113	0.2290	0.2197	0.2014	0.1332	0.1978	0.2023
trunc-4	0.2767	0.2682	0.2878	0.2704	0.2614	0.1918	0.2669	**0.2605**
trunc-5	0.2501	0.2410	0.2501	0.2461	0.2422	0.1551	0.2393	0.2320
Average	0.2269	**0.2278**	0.2251	0.2204	0.2150	0.1511	0.2086	

Table 2. MAP of different IR models and different stemmers for TD query formulation

	Mean Average Precision (TD)							
	Okapi	**DFR-I(n_e)C2**	**DFR-PL2**	**DFR-GL2**	**LM**	***tf idf***	**DFR-PB2**	**Average**
NoStem/NoSL	0.2360	0.2396	0.2303	0.2258	0.2367	0.1640	0.2375	0.2243
NoStem	0.2351	0.2381	0.2304	0.2239	0.2368	0.1639	0.2368	0.2236
Light Stem.	0.2351	0.2381	0.2304	0.2239	0.2368	0.1639	0.2368	0.2236
Aggressive	0.2351	0.2383	0.2304	0.2240	0.2368	0.1639	0.2375	0.2237
3-grams	0.2728	0.3121	0.2869	0.3163	0.2569	0.1856	0.0738	0.2435
4-grams	0.2632	0.2774	0.2619	0.2733	0.2436	0.1735	0.2558	0.2498
trunc-3	0.2800	0.2750	0.2743	0.2681	0.2640	0.1385	0.2531	0.2504
trunc-4	0.3226	0.3257	0.3381	0.3089	0.3215	0.2139	0.3186	**0.3070**
trunc-5	0.2928	0.2871	0.2869	0.2900	0.2938	0.1729	0.2758	0.2713
Average	0.2585	**0.2643**	0.2573	0.2547	0.2546	0.1698	0.2363	

Table 3. MAP of different IR models and different stemmers for TDN query formulation

	Mean Average Precision (TDN)							
	Okapi	**DFR-I(n_e)C2**	**DFR-PL2**	**DFR-GL2**	**LM**	***tf idf***	**DFR-PB2**	**Average**
NoStem/NoSL	0.2792	0.2638	0.2640	0.2729	0.2707	0.1789	0.2542	0.2548
NoStem	0.2800	0.2769	0.2753	0.2829	0.2824	0.1771	0.2686	0.2633
Light Stem.	0.2800	0.2769	0.2753	0.2829	0.2824	0.1771	0.2686	0.2633
Aggressive	0.2800	0.2765	0.2754	0.2832	0.2824	0.1771	0.2686	0.2633
3-grams	0.2597	0.3456	0.3019	0.2738	0.3114	0.2081	0.0232	0.2462
4-grams	0.2974	0.3169	0.3025	0.3004	0.2919	0.1874	0.3156	0.2874
trunc-3	0.3138	0.3204	0.3449	0.2835	0.3232	0.1478	0.2713	0.2864
trunc-4	0.3788	0.3768	0.3722	0.3801	0.3713	0.2261	0.3704	**0.3537**
trunc-5	0.3199	0.3188	0.3210	0.3075	0.3217	0.1842	0.3046	0.2968
Average	0.2953	**0.3024**	0.2985	0.2939	0.3002	0.1834	0.2620	

Referring to these results, a general overview of the issues that are addressed to analyze is as follows:

1. Comparing the performance of different IR models and discussing which retrieval model performs the best for different stemmers.
2. Between different stemming and indexing methods, which one is the most effective one and what are the reasons for the weak performance of certain strategies.
3. Whether applying stemming or indexing strategies has practically a better effect on performance than non-stemming methods.
4. Evaluating the *n*-grams and trunc-*n* indexing strategies to identify which value of *n* is the most appropriate one.
5. Verifying whether stopword removal helps to achieve a better retrieval performance comparing to approaches that ignore this operation.

5.2 IR Models Evaluation

Referring to Tables 1 through 3, we can see that DFR-I(n_e)C2 model has the best average performance for any given stemming or indexing strategy and any query formulation. This model is followed by Okapi and DFR-PL2 model for T and TD query formulations and by LM, DFR-PL2 and Okapi for TDN query formulation. We can also see that the classical *tf idf* vector space model has always the worst performance for any applied stemming and indexing strategy.

Although, we believe that in some cases the average measurements do not precisely describe the overall performance (e.g., it is known that extreme values influence the average). To overcome this inadequacy of average values, applying a query-by-query analysis could help to have a more precise understanding of the reasons behind the obtained results.

5.3 Query Formulation Evaluation

From the Tables 1, 2 and 3 we can see that expanding a query by adding the description and then the narrative logical sections improves the performance for any stemming and indexing strategy. To be more precise and considering the trunc-4 strategy as the best performing one, Table 4 shows the change in percentage when considering the TD and TDN query formulations. We can see that this query expansion makes a positive average improvement for all IR models. The results show that adding the description to the query makes an average improvement of +17.89% in performance while adding the narrative section as well as the description (TDN) improves the average performance for +35.78%.

Looking at the different IR models separately we can see that expanding the query has the most impact for LM model where the performance changes for +23.01% and +42.03% for TD and TDN respectively over T formulation.

Considering only the best performing model (DFR-I(n_e)C2) in our experiment we can see that the change percentage over T is +21.44% and +40.5% for TD and TDN respectively.

As for query formulation using only title (T) has obviously week retrieval performance, for the rest of our evaluation we consider only TD and TDN formulations. Also we omitted *tf idf* and DFR-PB2 IR models in further evaluations as according to the results they are clearly the models with the weakest performance.

Table 4. MAP of different query formulation (with trunc-4 and different IR models) & its change percentage for TD and TDN over T

	Mean Average Precision trunc-4				
	T	**TD**	**TDN**	**% of Change TD over T**	**% of Change TDN over T**
Okapi	0.2767	0.3226	0.3788	+16.62%	+36.90%
DFR-I(n_e)C2	0.2682	0.3257	0.3768	+21.44%	+40.50%

Table 4. (*continued*)

DFR-PL2	0.2878	0.3381	0.3722	+17.47%	+29.34%
DFR-GL2	0.2704	0.3089	0.3801	+14.23%	+40.58%
LM	0.2614	0.3215	0.3713	**+23.01%**	**+42.03%**
tf idf	0.1918	0.2139	0.2261	+11.51%	+17.89%
DFR-PB2	0.2669	0.3186	0.3704	+19.35%	+38.76%
Average	0.2605	0.3070	0.3537		
% of Change over T	base	+17.89%	+35.78%		

5.4 Stemming Strategies Evaluation

Tables 5 and 6 depict the MAP under different stemming strategies and stopword removal. The results show that removing the stopwords improves the average performance when using TDN query formulation while it does not have a positive impact when we have TD query formulation. This might be due to our stopword list which is quite a short list. As stopword removal plays an important role in retrieval effectiveness enhancement [19] it seems important to consider increasing the number of terms present in the stopword list.

Table 5. MAP of various IR models with different stemming strategies (TD query formulation)

	Mean Average precision (TD)			
	NoStem/NoSL	**NoStem**	**Light Stem**	**Aggressive**
Okapi	0.2360	0.2351	0.2351	0.2351
DFR-I(n_e)C2	0.2396	0.2381	0.2381	0.2383
DFR-PL2	0.2303	0.2304	0.2304	0.2304
DFR-GL2	0.2258	0.2239	0.2239	0.2240
LM	0.2367	0.2368	0.2368	0.2368
Average	0.2337	0.2329	0.2329	0.2329

Table 6. MAP of various IR models with different stemming strategies (TDN query formulation)

	Mean Average precision (TDN)			
	NoStem/NoSL	**NoStem**	**Light Stem**	**Aggressive**
Okapi	0.2792	0.2800	0.2800	0.2800
DFR-I(n_e)C2	0.2638	0.2769	0.2769	0.2765
DFR-PL2	0.2640	0.2753	0.2753	0.2754
DFR-GL2	0.2729	0.2829	0.2829	0.2832
LM	0.2707	0.2824	0.2824	0.2824
Average	0.2701	0.2795	0.2795	0.2795

The values in Tables 5 and 6 also show that applying either light or aggressive stemmers does not change the performance and the MAP stays almost the same with our without applying these stemmers. As Marathi language has a complex inflectional morphology the structure of applied stemmers should be reconsidered so that stemming would help to improve the performance. A more complex stemmer might help to augment the retrieval performance.

5.5 Indexing Strategies Evaluation

Referring to Tables 7 and 8 we will find that applying indexing strategies like *n*-grams or trunc-*n* clearly increases the retrieval performance comparing to no stemming method.

Table 7. MAP of various IR models with different indexing strategies (TD query formulation) and its change % over no-stemming approach

	Mean Average precision (TD)					
	NoStem	**3-gram**	**4-gram**	**trunc-3**	**trunc-4**	**trunc-5**
Okapi	0.2351	0.2728	0.2632	0.2800	0.3226	0.2928
DFR-I(n_e)C2	0.2381	0.3121	0.2774	0.2750	0.3257	0.2871
DFR-PL2	0.2304	0.2869	0.2619	0.2743	0.3381	0.2869
DFR-GL2	0.2239	0.3163	0.2733	0.2681	0.3089	0.2900
LM	0.2368	0.2569	0.2436	0.2640	0.3215	0.2938
Average	0.2329	0.2890	0.2639	0.2723	0.3234	0.2901
% of Change over base	base	+24.1%	+13.3%	+16.9%	**+38.9%**	+24.6%

Table 8. MAP of various IR models with different indexing strategies (TDN query formulation) and its change % over no-stemming approach

	Mean Average precision (TDN)					
	NoStem	**3-grams**	**4-grams**	**trunc-3**	**trunc-4**	**trunc-5**
Okapi	0.2800	0.2597	0.2974	0.3138	0.3788	0.3199
DFR-I(n_e)C2	0.2769	0.3456	0.3169	0.3204	0.3768	0.3188
DFR-PL2	0.2753	0.3019	0.3025	0.3449	0.3722	0.3210
DFR-GL2	0.2829	0.2738	0.3004	0.2835	0.3801	0.3075
LM	0.2824	0.3114	0.2919	0.3232	0.3713	0.3217
Average	0.2795	0.2985	0.3018	0.3171	0.3758	0.3178
% of Change over base	base	+6.8%	+8.0%	+13.5%	**+34.5%**	+13.7%

From the obtained values we can see that between n-grams and trunc-n strategies (with different values for n), trunc-4 method clearly increases the mean performance the most and gives, almost always, the best performance (except for DFR-GL2 model in TD query formulation where 3-gram gives a better result).

We can say that for the trunc-n approach the best value for n is 4. For the n-gram models, 3-gram tends to have a better performance than 4-gram. While both language-independent indexing strategies increase the performance comparing to no stemming approach and even stemming approach (applying the proposed light and aggressive stemmers).

5.6 Pseudo-Relevance Feedback

According to our previous experiments with different languages we see that applying a blind-query expansion (or pseudo-relevance feedback) (PRF) might help to improve the mean retrieval effectiveness.

In this expansion the original query is reformulated by adding m terms extracted from the k top ranked documents. In this experiment, we applied Rocchio's approach [20] with $\alpha = 0.75$, $\beta = 0.75$. The expansion is applied to both TD and TDN query formulations. The results are shown in Table 9 and Table 10.

Table 9. MAP of Different Blind-Query Expansions, Rocchio's method, TD queries

	Mean Average Precision (TD)		
	NoStem	**3-gram**	**trunc-3**
Okapi	0.2351	0.2728	0.2800
3 docs / 20 terms	0.2372	0.2944	0.2833
3 docs / 50 terms	0.2406	0.2849	**0.3122**
3 docs / 70 terms	0.2421	0.2860	0.3088
3 docs / 100 terms	0.2335	0.2889	0.3004
3 docs / 150 terms	0.2258	0.2804	0.2788
5 docs / 20 terms	0.2406	0.2957	0.2878
5 docs / 50 terms	0.2144	0.2892	0.2856
5 docs / 70 terms	0.2440	0.2913	0.2874
5 docs / 100 terms	0.2382	0.2882	0.3036
5 docs / 150 terms	0.2304	0.2899	0.2960
10 docs / 20 terms	0.2407	0.2967	0.2894
10 docs / 50 terms	0.2431	0.2874	0.2896
10 docs / 70 terms	0.2465	0.2874	0.3064
10 docs / 100 terms	0.2446	0.2938	0.3050
10 docs / 150 terms	0.2442	0.2932	0.2954
15 docs / 20 terms	0.2349	0.2972	0.2853
15 docs / 50 terms	0.2440	0.2908	0.2894
15 docs / 70 terms	0.2454	0.2882	0.3046
15 docs / 100 terms	0.2478	0.2922	0.3003
15 docs / 150 terms	0.2468	0.2954	0.2967

Table 10. MAP of Different Blind-Query Expansions, Rocchio's method, TDN queries

	Mean Average Precision (TDN)		
	NoStem	**3-gram**	**trunc-3**
Okapi	0.2800	0.2597	0.3138
3 docs / 20 terms	0.2614	0.2944	0.3239
3 docs / 50 terms	0.2736	0.3231	0.3290
3 docs / 70 terms	0.2838	0.3325	0.3279
3 docs / 100 terms	0.2810	0.3454	0.3269
3 docs / 150 terms	0.2685	**0.3461**	0.3218
5 docs / 20 terms	0.2613	0.2943	0.3239
5 docs / 50 terms	0.2718	0.3203	0.3322
5 docs / 70 terms	0.2725	0.3393	0.3310
5 docs / 100 terms	0.2808	0.3354	0.3314
5 docs / 150 terms	0.2786	0.3414	0.3279
10 docs / 20 terms	0.2622	0.2900	0.3292
10 docs / 50 terms	0.2700	0.3179	0.3358
10 docs / 70 terms	0.2731	0.3392	0.3336
10 docs / 100 terms	0.2738	0.3383	0.3310
10 docs / 150 terms	0.2790	0.3413	0.3303
15 docs / 20 terms	0.2618	0.2957	0.3271
15 docs / 50 terms	0.2742	0.3188	0.3311
15 docs / 70 terms	0.2753	0.3421	0.3265
15 docs / 100 terms	0.2763	0.3382	0.3229
15 docs / 150 terms	0.2757	0.3404	0.3192

In our experiment the results show that applying the pseudo-relevance feedback approach enhance the mean retrieval performance. The best results were gained for the Okapi model.

When using TD query formulation (see Table 9), the best enhancement is for the trunc-3 model when the queries are expanded adding 50 terms selected from the first 3 retrieved documents. Here the MAP is increased for +11.5% (from 0.2800 to 0.3122).

When it comes to TDN query formulation (see Table 10) the enhancement of performance becomes even bigger. Here for the best case the MAP is increased for +33.30% (from 0.2597 to 0.3461) for 3-grams strategy. As Table 10 shows for TDN query formulation the blind-query expansion improves a lot the retrieval effectiveness for Okapi model using 3-grams indexing strategy.

The results show that adding more than 100 terms does not help any better the enhancement. We believe that this is due to the fact that including more terms causes noises for the system [21].

6 Conclusion

The results of our experiment in FIRE 2011 evaluation campaign show that in general the IR models DFR-I(n_e)C2 and DFR-PL2, both based on Divergence From Randomness paradigm, are giving the best retrieval results for any stemming or indexing strategies. These models are followed by Okapi model. The classical *tf idf* model tends to offer lower performance levels.

The results also show that in general expanding the query by adding the description (D) and narrative (N) sections to it improves the retrieval effectiveness comparing to using only the title (T) part of the query. For the best performing model (DFR-I(n_e)C2), enlarging the query from T to TD improves the retrieval effectiveness up to +21.41%. This improvement increases to +40.5% when changing the query formulation to TDN. In average for all seven models there were between +17.89% and +35.78% enhancement in performance while using TD and TDN respectively (over Title only formulation).

We can see from the results that the light and aggressive stemmers, proposed in this experiment, did not change the performance comparing to no stemming approach. But applying *n*-gram or trunc-*n* indexing strategies clearly increases the retrieval performance comparing to no stemming method. In our experiment trunc-4 approach tends to result the best MAP.

The results after applying the Rochio's approach as the adopted approach for blind-query expansion show that this expansion tends to help the retrieval enhancement. Here the blind-query expansion increases the retrieval effectiveness the most for trunc-3 and 3-gram strategies while using the Okapi IR model.

Acknowledgements. This work was supported in part by the Swiss National Science Foundation under Grant #200020-129535/1.

References

1. Dolamic, L., Savoy, J.: UniNE at FIRE 2008: Hindi, Marathi and Bengali IR. FIRE 2008 Working Notes (2008)
2. Savoy, J.: Combining Multiple Strategies for Effective Monolingual and Cross-Lingual Retrieval. IR Journal 7, 121–148 (2004)
3. Savoy, J.: Comparative Study of Monolingual and Multilingual Search Models for Use with Asian Languages. ACM - Transactions on Asian Languages Information Processing 4, 163–189 (2005)
4. Savoy, J.: Searching Strategies for the Hungarian Language. Information Processing & Management 44(1), 310–324 (2008)
5. Koskenniemi, K., Church, K.W.: Complexity Two-Level Morphology and Finnish. In: Proceedings COLING, Budapest, pp. 1–9 (1988)
6. Voorhees, E.M., Harman, D.K. (eds.): TREC. Experiment and Evaluation in Information Retrieval. The MIT Press, Cambridge (2005)
7. Robertson, S.E., Walker, S., Beaulieu, M.: Experimentation as a Way of Life: Okapi at TREC. Information Processing & Management 36, 95–108 (2002)

8. Amati, G., van Rijsbergen, C.J.: Probabilistic Models of Information Retrieval Based on Measuring the Divergence from Randomness. ACM Transactions on Information Systems 20, 357–389 (2002)
9. Hiemstra, D.: Using Language Models for Information Retrieval. Ph.D. Thesis (2000)
10. Hiemstra, D.: Term-Specific Smoothing for the Language Modeling Approach to Information Retrieval. In: Proceedings of the ACM-SIGIR, pp. 35–41. The ACM Press (2002)
11. Zhai, C., Lafferty, J.: A Study of Smoothing Methods for Language Models Applied to Information Retrieval. ACM Transactions on Information Systems 22, 179–214 (2004)
12. Fox, C.: A Stop List for General Text. ACM-SIGIR Forum 24, 19–35 (1990)
13. McNamee, P., Mayfield, J.: Character N-gram Tokenization for European Language Text Retrieval. IR Journal 7, 73–97 (2004)
14. Savoy, J.: Light Stemming Approaches for the French, Portuguese, German and Hungarian Languages. In: Proceedings of the ACM-SAC, pp. 1031–1035. The ACM Press (2006)
15. Harman, D.K.: How Effective is Suffixing? Journal of the American Society for Information Science 42, 7–15 (1991)
16. Porter, M.F.: An Algorithm for Suffix Stripping. Program 14, 130–137 (1980)
17. Fautsch, C., Savoy, J.: Algorithmic Stemmers or Morphological Analysis: An Evaluation. Journal of the American Society for Information Sciences and Technology 60, 1616–1624 (2009)
18. Buckley, C., Voorhees, E.M.: Retrieval System Evaluation. In: Voorhees, E.M., Harman, D.K. (eds.) TREC. Experiment and Evaluation in Information Retrieval, pp. 53–75. The MIT Press, Cambridge (2005)
19. Dolamic, L., Savoy, J.: When Stopword Lists Make the Difference. Journal of the American Society for Information Sciences and Technology 61, 200–203 (2010)
20. Buckley, C., Singhal, A., Mitra, M., Salton, G.: New Retrieval Approaches Using SMART. In: Proceedings of the TREC-4, pp. 25–48. NIST Publication #500-236, Gaithersburg (1996)
21. Peat, H.J., Willett, P.: The Limitations of Term Co-Occurrence Data for Query Expansion in Document Retrieval Systems. Journal of the American Society for Information Science 42, 378–383 (1991)