Retrieval effectiveness study with Farsi language

Mitra Akasereh, Jacques Savoy

Université de Neuchâtel
Institut d'informatique - Rue Emile-Argand 11, CH - 2000 Neuchâtel
Mitra.Akasereh@unine.ch, Jacques.Savoy@unine.ch

ABSTRACT. *Having Farsi as the underlying language and using a test collection of 166,774 documents and 100 topics, this experiment evaluates the retrieval effectiveness of different IR models while using a light and a plural stemmer as well as n-grams and trunc-n indexing strategies. Moreover the impact of stoplist removal is evaluated. According to the obtained results the DFR-I(n_e)C2 model is the best performing one. The proposed light and plural stemmer improve the retrieval performance compare to non-stemming approach. Indexing strategies trunc-4 and trunc-5 have also a positive impact on the performance while 3-grams and trunc-3 have the most negative impact on the results. The results reveal that for Farsi stoplist removal plays an important role in improving the retrieval performance. A query-by-query analysis on the results shows that avoiding extreme results would be possible by adding extra controls and rules, according to Farsi morphology, to the stemming algorithms.*

RÉSUMÉ. *Dans le but d'utiliser le persan comme langue de référence, et en utilisant une collection test de 166 774 documents et de 100 requêtes, cette étude évalue la performance des différents modèles de RI sur lesquels sont appliquées diverses stratégies d'indexation et de recherche. De plus, cette étude évalue l'impact de l'élimination de la liste des mots-outils lors de l'indexation. Selon les résultats obtenus, le modèle DFR-I(n_e)C2 est le plus performant. L'enracineur léger et l'enracineur pluriel améliorent la performance en comparaison à l'approche sans enracineur. Les stratégies d'indexation, comme tronc-4 et tronc-5 améliorent la performance, alors que les approches comme 3-grams et tronc-3 ont l'impact le plus négatif sur les résultats. Les résultats révèlent que l'élimination de la liste des mots-outils joue un rôle important dans l'amélioration de la performance. L'analyse requêtes par requêtes montre qu'il serait possible d'ajouter des règles supplémentaires aux enracineurs, pour éviter des résultats erronés.*

KEY WORDS: *Farsi language, Information retrieval in Persian, Persian morphology*

MOTS-CLÉS: *Langue persane, Recherche d'information en persan, morphologie du persan*

1. Introduction

In spite of the constant growing of the need for information retrieval (IR) tools dealing with languages other than English, there has been much less done for Persian language (Farsi). In order to create effective IR tools for a new language a good basis would be readapting portions of certain existing retrieval systems for this new language (Savoy 1999). Of course restructuring these existing tools is not a trivial task. Different languages with different linguistic characteristics have their particular affects and restrictions in the process of IR systems development. A good start point is to choose a proper IR model and then to provide the necessary linguistic tools considering the characteristics (e.g., grammatical, morphological) of the target language (Savoy, 2004).

Accordingly in our experiment studying Farsi language, we first study Farsi morphology. We then propose a light and a plural (very light) stemmer along with a stopword list for this language. And finally after applying these with different IR models on our test collection, we make a query-by-query analysis on the results to discover the weaknesses of these different methods. Our aim is to then create more accurate and effective IR tools for this language for both monolingual and bilingual retrieval.

The rest of this paper is organized as follows: Section 2 is a brief introduction to Persian language and its morphology. Section 3 represents the setup of our experiment. Section 4 contains the obtained results and states the related analysis while Section 5 concludes the experiment.

2. Persian language

Persian language, also known as Farsi, is a subclass of the western Iranian languages. This language is a member of Indo-European languages and in terms of orthography it belongs to the Arabic script-based languages. The underlying morphology is a bit more complex than English but it is not a difficult one compared to languages such as Turkish or Finnish (Dolamic & Savoy, 2009). Having more than 100 million native speakers Persian language is the official language of Iran, Afghanistan and Tajikistan (called “Persian or Farsi”, “Dari” and “Tajik” respectively). It is also spoken in parts of some other countries in the Middle East. As a member of Indo-European languages, Persian is to some extends related to the majority of European languages, including English and German, and has been in interaction with other non-Iranian languages like Arabic, Turkish, Hindi, Mongolian, Latin, Greek, Russian and French (Dolamic & Savoy, 2009; Bijankhan *et al*., 2011).

2.1. *Persian morphology & problems with Persian language*

Persian language is written using 32 letters: the 28 Arabic letters plus four more characters, "گ" /g/, "چ" /tʃ/, "پ" /p/, "ژ" /ʒ/, which are not used in classical Arabic. These letters are written from right to left, and for many of them the form differs if they are connected to another letter or isolated, at the beginning, in the middle or as the final letter in a word.

2.1.1. *Word segmentation*

Defining the word's boundary in Persian is, sometimes, a challenging task. For some cases there is not a unique way of writing which makes it difficult to distinguish the word's boundary. The cursive nature of Arabic script causes this problem. Words which consist of minimum two morphemes can be written either concatenated or separately. Only if the first morpheme ends with one of the letters that cannot be concatenated to the next letter ("ا" or "آ" /ɒ:/, "ـد" /d/, "ـذ" /z/, "ـر" /r/, "ز" /z/, "ژ" /ʒ/, "ـو" /v/) then the two morphemes cannot be concatenated. But still when morphemes are not concatenated, there would be two different ways to write them: with a blank space between two morphemes or the Zero-Width-Non-Joiner (ZWNJ) (represented by # in the following example). In general for a word made of *n* morphemes there are at most 3^{n-1} different ways of writing among which some of them are not correct or less common (indicated by * in the example below). For example having the word "abc", in which *n*=3 and considering that morphemes "a" and "b" end with a concatenative letter there will be 9 possible forms of writing:
{abc*, a b c, a#b#c, ab#c, ab c, a#bc*, a bc*, a b#c, and a#b c}. Although there are orthographic rules, published by the Persian Academy of Language and Literature (PALL), for the grammar of Persian orthography still there exists orthographic variation in different texts (Bijankhan *et al.,* 2011). This variety of writing creates challenges in the process of tokenization as well as stemming phase.

2.1.2. *Inflectional morphology*

Like other Indo-European languages Persian has affixitive morphology. This means that words are modified by concatenating suffixes or prefixes to them (Dolamic & Savoy*,* 2009). Adding a prefix or suffix to a stem may cause a change in the word's orthography which makes the process of stemming and lemmatization more complex. Therefore extra rules are needed to conflate related words to their common stem (some examples of this phenomenon are mentioned in the next paragraphs).

In Persian there are different ways to define different grammatical cases. The genitive and possessive cases can be shown by coupling two nouns using the particle "◌ِ" /-e/ known as "ezafe". Ezafe usually in not written and only pronounced. This particle changes to a "ء" /ʔ/, known as "hamze", if the noun ends by an attached "ه" /h/ written as "ـهٔ" /-je/ and to a "ی" /-je/ if the word ends with "ا" /ɒ:/ or "و" /u:/. These are cases where "ezafe" is appeared in writing. The possessive construction as explained is for cases where the possessor is mentioned after the object if not the possessive case is shown by adding different suffixes regarding the person of the possessor (i.e., “ام” /-æm/, “یت” /-et/, “یش” /-esh/, “مان” /-mɒn/, “تان” /-tɒn/, “شان” /-ʃɒn/).

To denote the plural, different suffixes can be used in Persian. The suffix “ها” /-hɒ/ is reserved for non-human nouns. This suffix can be written both attached to the word or separately. The suffix “ان” /-ɒn/ is used for human nouns and becomes “گان” /-gɒn/ if the word ends by “ه” /h/ and sometimes changes to “یان” /-jɒn/ if the word ends by other vowels. For nouns borrowed from Arabic, the plural form can be formed either by adding the suffix “ات” /-ɒt/ or the suffix “ین” /-in/ or the broken form for plurals, which owns an irregular format. While making plural form using suffix “گان” /-gɒn/ if the word ends with a silent “ه” /he/ this letter will be omitted from the end of the word. The suffix then either attaches to the word or stays separately if the last letter (after omitting the “ه” /he/) is among the letters that cannot be joined to next letter.

In Persian comparatives are formed by adding the suffix tar “تر” /-tær/ (with some exceptional cases where the word completely changes). The superlatives are built by adding the suffix “ترین” /-tærin/. And to make relative adjectives there are different suffixes most commonly the suffix “ی” /-je/. In other cases suffixes “ه” /h/, “ین” /-in/ and rarely “گان” /-gɒn/ can be found.

There is no grammatical gender in Persian language. In order to specify the natural gender the words “man” and “woman” or the adjectives “male” and “female” are using.

2.1.3. *Other grammatical issues*

Persian language does not have specific definite or indefinite articles (as “the” and “a” or “an” in English). In general definite or indefinite words can be distinguished by looking at the structure of a phrase rather than the existence of a specific particle. Indefinite articles can be expressed in two ways. First with the suffix “ی” /-je/ which changes to “ای” /-i/ if the word ends with a silent “ه” /h/. It changes to “یی” /-jiː/ while being added to a plural noun made by adding the suffix “ها” /-hɒ/. Second, by adding the numeral “یک” /jek/ (one) before the expected noun. For the definite nouns the particle “را” /ɾɒ/ (the particle following the noun in accusative cases) is considered to have the function of the definite article. The relative “ی” /-je/ as explained above can also have the function of definite article if the word, to which suffix “ی” /-je/ is added followed by the particle “که” /ke/. While in spoken grammar adding an ending “ه” /h/ to the word makes the word definite.

In Farsi proper names are written in the same way as other words so it is not easy to prevent them from being stemmed. For example the word “ایران” (Iran) could be stemmed into “ایر” as the suffix “ان” /-ɒn/ is one of the suffixes used for pluralisation.

In this language homonyms exist as in all other languages. But additionally the fact that the short vowels /æ/, /e/, /o/ are not written in Persian script, causes a remarkable number of double or even triple heteronyms and thus resulting ambiguity (e.g., “سر” /seɾ/, “secret”, “سر” /sæɾ/, “head” and “سر” /soɾ/, “slippery”) (AleAhmad *et al.,* 2008).

3. Experiment architecture

3.1. *Test collection*

The test-collection used for this experiment is the collection made available during the CLEF 2008. This collection is made up of 166,774 newspaper articles, with approximately 202 terms per document (after stopword removal), extracted from a national Iranian newspaper (“Hamshahri”) between the years 1996 to 2002. There are 100 topics (from Topic #551 to Topic #650) in the collection, where the first 50 topics are made and assessed during CLEF2008 and the last 50 ones during CLEF2009. The topics have a total number of 9,625 relevant documents with mean of 96.25, median of 89.5 (standard deviation 62.14). The topics cover a variety of subjects such as politics, literature, art, economics, etc. in both national and international domain. Subjects like “Electronic commerce”, “smoking & heart disease”, “cinema”, “Tehran international book fair”, “Football world cup” or “Global oil price variation”. Topic #574 (“قهرمان لیگ برتر”, “Champion of first Pro League”) with 7 relevant items has the smallest number of pertinent documents while Topic #649 (“بحران نفت دولت خاتمی”, “Khatami government oil crises”) with 266 relevant items has the greatest number of relevant documents. Following the TREC model, 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. Although in this experiment only the “title” section is used. The corpus is coded in UTF-8.

3.2. *Stoplist, stemming and indexing strategies*

The applied stopwords list contains 881 terms covering the frequent terms such as determinants, prepositions, conjunctions, pronouns, different forms of some auxiliary verbs and also some suffixes (for cases where suffixes are written separately from the word).

As indexing strategies different automatic indexing methods is applied on the collection to be evaluated and compared. Two, language-independent, indexing approaches that are used are *n*-grams (which is the act of producing the overlapping sequences of n characters (McNamee & Mayfield, 2004) and trunc-*n* (which is the process of truncating a word by keeping its first *n* characters and cutting of the remaining letters). Different values for n, for both *n*-grams and trunc-*n* are tested to find which value of *n* gives the best performance.

For stemming a light suffix-stripping algorithm is used which removes the morphological suffixes (mostly inflections) such as possessive, plural, relative, etc. The removal is adjusted by quantitative restrictions, means that for the removal the length of the term is taken into consideration in order to have a meaningful sequence as the result and also not to remove a whole word entirely. The procedure is mostly focused on nouns and adjectives and different forms of verbs are not taken into account. Another stemmer is also tested on the collection that removes only the plural suffixes. The proposed stoplist and light stemmer are freely available at www.unine.ch/info/clef/.

3.3. *IR models*

Six different IR models are implemented in the experiment to be evaluated and compared. The models are as follows:

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 D_j (tf_{ij}) and the logarithm of its inverse document frequency (idf_j). We normalized the index weights using cosine normalization (Manning *et al.,* 2008).

As another vector-space model, the Lnu-ltc model, suggested by (Singhal, 2002), is adopted. In this model the length of the document is taken into account. Here the index weight for the document term (Lnu) is calculated as:

$$w_{ij} = [\log(tf_{ij})+1] \,.\, norm_i \qquad [1]$$

$$\text{with} \quad norm_i = \frac{1}{(1+\log(\frac{\sum tf_{ij}}{nt_i}))\cdot((1-slope)\cdot pivot + (slope\cdot nt_i))}$$

Where nt_i is the length of document D_i (number of its index terms), *slope* and *pivot* are constants. The index weight for the query term (ltc) is calculated as:

$$w_{qj} = (\log(tf_{qj})+1)\cdot idf_j \cdot norm_q \qquad \text{with} \quad norm_q = \frac{1}{\sqrt{\sum_k (tf_{qk}\cdot idf_j)^2}} \qquad [2]$$

As the first probabilistic model the Okapi (BM25) (Robertson *et al.,* 2000) model, which also takes the document into account is used. For this model the parameters are fixed as b=0.75, k_1=1.2 and *advl*=202.

Also two other probabilistic models, DFR-PL2 and DFR-I(n_e)C2 based on measuring the divergence from randomness (DFR) family are used (Amati *et al.,* 2002). Here we have:

$$w_{ij} = Inf_{ij}^{1} \cdot Inf_{ij}^{2} = -\log_2(\mathrm{Prob}_{ij}^{1}(tf_{ij})) \cdot (1 - \mathrm{Prob}_{ij}^{2}(tf_{ij})) \qquad [3]$$

DFR-PL2 is defined by:

$$\mathrm{Prob}_{ij}^{1} = \frac{e^{-\lambda_j} \cdot \lambda_j^{tfn_{ij}}}{tfn_{ij}!} \qquad [4]$$

$$\mathrm{Prob}_{ij}^{2} = \frac{tfn_{ij}}{tfn_{ij}+1} \quad \text{with} \quad tfn_{ij} = tf_{ij} \cdot \log_2(1 + \frac{c \cdot mean\_dl}{l_i}) \qquad [5]$$

And DFR- I(n_e)C2 is defined by:

$$Inf_{ij}^{1} = tfn_{ij} \cdot \log\left[\frac{n+1}{n_e + 0.5}\right] \quad \text{with} \quad n_e = n \cdot (1 - (\frac{n-1}{n})^{tc_j}) \qquad [6]$$

$$\mathrm{Prob}_{ij}^{2} = 1 - \frac{tc_j + 1}{df_j \cdot (tfn_{ij} + 1)} \qquad [7]$$

And finally one approach based on language model (LM) known as non-parametric probabilistic model is employed. Here the adopted model is the one suggested by Hiemstra (Hiemstra, 2000), which defined as follow where λ_j is a smoothing factor (set to 0.35 for all index terms) and lc is an estimation of the corpus C length:

$$P(d_i/q) = P(d_i) \cdot \prod_{t_j \in q} \left\lfloor \lambda_j \cdot P(t_j/d_i) + (1-\lambda_j) \cdot P(t_j/C) \right\rfloor \qquad [8]$$

with $P(t_j/d_i) = \frac{tf_{ij}}{l_i}$ and $P(t_j/C) = \frac{df_j}{lc}$ ($lc = \sum_k df_k$)

3.4. *Evaluation*

To evaluate the retrieval performance the mean average precision (MAP) is used based on the 100 queries. The usage of mean provides the same level of importance for all the queries. In some cases the average measurements do not sufficiently describe the total performance (e.g., when extreme values influence the average). To overcome this inadequacy a query-by-query analysis is also applied for some of the models and strategies. Looking at specific examples helps to have a more precise understanding of the reasons behind the obtained results and gather more detailed information on how different strategies work.

4. Results and analysis

Table 1 shows the results obtained during the experiment. In the following sections, referring to these results, different aspects are addressed to analyse.

Table 1. Mean average precision (MAP) of different IR models and different stemmers

	Mean Average Precision					
	LM	**DFR-PL2**	***tf idf***	**DFR-I(n_e)C2**	**Okapi**	**Lnu-ltc**
no stem. /no stoplist	0.3449	0.3905	0.2156	0.4087	0.3815	0.3729
no stem.	0.3592	0.4025	0.2648	0.4069	0.3962	0.3763
3-grams	0.3212	0.3743	0.2173	0.3982	0.3563	0.3507
4-grams	0.3325	0.3770	0.2499	0.4060	0.3916	0.3574
5-grams	0.3463	0.3850	0.2581	0.4068	0.3911	0.3601
6-grams	0.3580	0.3963	0.2607	0.4091	0.3959	0.3686
light stemmer	0.3668	0.4155	0.2599	0.4168	0.4076	0.3874
light stem. /no stoplist	0.3433	0.3982	0.2040	0.4117	0.3785	0.3737
plural stemmer	0.3636	0.4082	0.2696	0.4124	0.4010	0.3806
trunc-3	0.3402	0.4000	0.2139	0.3955	0.3870	0.3619
trunc-4	0.3635	0.4186	0.2584	**0.4189**	0.4084	0.3862
trunc-5	0.3676	0.4148	0.2687	0.4185	0.4077	0.3859
Average	0.3506	0.3984	0.2451	0.4091	0.3919	0.3718

4.1. *IR Models Evaluation*

Referring to Table 1 the best performing IR model is DFR-I(n_e)C2 for any given stemming or indexing strategy. After DFR-I(n_e)C2 model the best overall performances are respectively for DFR-PL2 and Okapi models.

When comparing DFR-I(n_e)C2 model with Okapi model (which has also good MAP results), by applying trunc-4 strategy, DFR-I(n_e)C2 model performs better for 72 queries out of 100 (producing a higher AP). But the difference between APs for each topic is not remarkable. The biggest difference is seen for Topic #594 (" قیمت بلیط هواپیما", "Flight prices") for which DFR-I(n_e)C2 model performs an AP of 0.3256 while Okapi gives 0.2092. In general number of topics for which the change is bigger than 15% is only 11.

Having DFR-I(n_e)C2 scheme as the model with the best performance, for this model the best stemming or indexing strategies are respectively trunc-4, light stemmer and trunc-5 (with a small difference between each of them). This is the case for almost all the evaluated models except for the classical *tf idf* model for which plural stemmer and trunc-5 perform the best.

One reason for which the truncating methods, with value of n equal to 4 or 5, performances are approximately the same as light stemmer can be explained as follow: Persian words are normally short terms so cutting off the end of a word after four or five characters either does not change the word at all (the word stays as it is) or leads to cutting off only the suffixes if there is any suffix attached to the term. Performing a query-by-query analysis confirms this conclusion. Comparing the performance of these three methods for the 100 queries using DFR-I(n_e)C2 model shows that there is not a big difference between the performances for each query except in 3 or 4 cases. One of these extreme cases is found for Topic #554 (“ سلامتی و استرس” ،“health & stress”), here the worth performance is for trunc-4 strategy (AP of 0.0301 compare to 0.3304 for trunc-5 and 0.3242 for light stemmer) this is due to the fact that by truncating the two terms of the topic, the result (“استر” & “سلام”) causes ambiguity so there are documents with high ranks which are not really relevant to the subject. Another extreme case is for Topic #630 (“جشنهای سنتی ایرانی” ، “Iranian traditional celebrations”) for which the light stemmer is the best performing one. With the trunc-4 or trunc-5 strategies the plural and genitive suffixes in (“جشنهای”, “celebrations”) are not removed properly. The same problem occurs for relative suffix “ی” in the term (“سنتی”, “traditional”) which still remains attached to the term by applying trunc-4 or trunc-5. But with the light stemmer these suffixes plus the relative suffix “ی” in (“ایرانی”, “Iranian”) will be deleted correctly returning the correct stem and consequently retrieving the documents with any different forms for the stems (“جشن” “سنت” & “ایران” ،“celebration”, “tradition” & “Iran”). As a result an AP of 0.4076 is obtained with light stemmer compare to 0.2255 and 0.2253 with trunc-5 and trunc-4. Same case is for Topic #648 (“حمله برجهای دوقلو”, “Twin towers attack”) where by applying the light stemmer the plural and genitive suffixes are deleted from the word “برجهای” and it is correctly indexed under its lemma “برج”. This was not the case when applying trunc-4 or trunc-5 methods. For Topic #650 (“نوسانات بنزین وارداتی”, “Imported fuel price volatility”) we have Aps of the best AP is 0.4692 for trunc-5, 0.3379 for trunc-4 compare to 0.1885 for light stemmer. Hear the weak performance of light stemmer is due to over-stemming of the term “بنزین” which is stemmed into “بنز” and causes retrieving of non-relevant documents with higher ranks than relevant ones.

4.2. *Differences between stemming and non-stemming approaches*

Table 2 shows for each stemming strategy the average of its performance for the six different IR models as well as the change percentage of this average over the average performance of no stemming approach.

Based on these values the trunc-5 strategy has the best average of 0.3772 and an improvement of 2.6%. With a small difference trunc-4 and light stemmer are the next best ones having an improvement of 2.2%. Applying methods like *n*-grams (especially with a small value of *n*) or trunc-3 clearly decrease the retrieval performance comparing to no stemming method. It can also be driven from the results that applying approaches like 3-grams or trunc-3 having more negative impact on a simple model like *tf idf* than on a robust one like DFR-I(n_e)C2. The change percentage of MAP using 3-grams over no stemming approach for *tf idf* model is -17.9% compare to -2.1% for DFR-I(n_e)C2. And by using trunc-3 this percentage is -19.2% for *tf idf* compare to -2.8% for DFR-I(n_e)C2.

Table 2. Average performance for stemming strategies & its change percentage over no stemming approach

	Mean Average Precision							
	LM	**DFR-PL2**	***tf idf***	**DFR-I(n_e)C2**	**Okapi**	**Lnu-ltc**	**Average**	**Change % over no stem.**
no stem.	0.3592	0.4025	0.2648	0.4069	0.3962	0.3763	0.3677	
3-grams	0.3212	0.3743	0.2173	0.3982	0.3563	0.3507	0.3363	-8.5%
4-grams	0.3325	0.3770	0.2499	0.4060	0.3916	0.3574	0.3524	-4.1%
5-grams	0.3463	0.3850	0.2581	0.4068	0.3911	0.3601	0.3579	-2.7%
6-grams	0.3580	0.3963	0.2607	0.4091	0.3959	0.3686	0.3648	-0.8%
light stem.	0.3668	0.4155	0.2599	0.4168	0.4076	0.3874	0.3757	+2.2%
Pl. stem.	0.3636	0.4082	0.2696	0.4124	0.4010	0.3806	0.3726	+1.3%
trunc-3	0.3402	0.4000	0.2139	0.3955	0.3870	0.3619	0.3498	-4.9%
trunc-4	0.3635	0.4186	0.2584	0.4189	0.4084	0.3862	0.3757	+2.2%
trunc-5	0.3676	0.4148	0.2687	0.4185	0.4077	0.3859	0.3772	+2.6%

Considering the best performing model (DFR-I(n_e)C2), the MAP results for this model applying different strategies and their change percentage over no stemming method is shown separately in Table 3. The results shows that again the trunc-4 and trunc-5 strategies have the most improvement over no stemming and trunc-3 and 3-grams decrease the performance comparing to no stemming approach.

Table 3. MAP for DFR-I(n_e)C2 model and different stemming methods & change percentage over no stemming approach

	no stem.	**3gram**	**4gram**	**5gram**	**6gram**	**light stem.**	**plural stem.**	**trunc3**	**trunc4**	**trunc5**
MAP for DFR-I(n_e)C2	0.407	0.398	0.406	0.407	0.409	0.417	0.412	0.396	0.419	0.419
Change % over no stem.		-2.2%	-0.2%	-0.0%	+0.5%	2.4%	+1.4%	-2.8%	+2.9%	+2.8%

A query-by-query analysis on this model shows that among the 100 queries trunc-4 improves the average precision for 56 queries comparing to no stemming while no stemming gives a better AP for 63 queries comparing to the results of trunc-3. This analysis also shows that in the cases where stemming strategies give a better result than no stemming approach (strategies with positive change percentages in Table 3), the number of queries for which the AP is improved is more or less equal to which the value of AP decreased. Comparing with no stemming: 6-grams method improves the AP for 50 topics, light stemmer for 51, plural stemmer for 43 and trunc-5 for 47.

Results in Tables 2 and 3 reveal that even though light stemmer and plural stemmer approaches give better performance that no stemming approach but the difference is rather limited. One reason can be the fact that for many topics suffixes in are written separately from the words so in these cases the stemming algorithm will not transform the word. Thus the result will be the same as ignoring the stemming phase. For the same reason by looking at results in Table 2 we can see that there is not a notable difference between the MAP obtained with light stemmer and the plural stemmer. Because in the former the plural suffixes are removed by the stemmer and for other kinds of suffixes, if they are detached from the term, the stoplist removal will do the deletion. For example in Topics #569 (“ پرونده های فساد اقتصادی” “Economic corruption cases”) or #617 (“جاذبه های گردشگری” “Tourist attractions”) where the relative plural and genitive suffixes “ی” “ها” and “ی” are separated from related terms the AP resulted from light stemmer does not differ from the one resulted with no stemming approach. On the contrary, by looking at topics where suffixes are attached to the terms the efficiency of light stemmer can be noticed. For Topic #630 (“جشنهای سنتی ایرانی” ،“Iranian traditional celebrations”), where the same suffixes are attached to the searched keywords, the AP with DFR-I(n_e)C2 model without stemming is 0.2981. This performance rises to 0.5858 by applying the light stemmer. Another example is Topic #648 (“حمله برجهای دوقلو”, “Twin towers attack”), where again the plural suffixe “ها” is concatenated to the term “tower”. In this example, the AP raises from 0.1267 (without stemming) to 0.2729 after applying a light stemmer. Another reason is over-stemming. In topic #600 (“تورم در ایران” “Inflation in Iran”) the term “تورم” “inflation” changes to “تور” after stemming which is the same stem for the term “tour”. As a result among the first 10 documents retrieved by applying light stemmer, documents talking about traveling tours can be found which results a decrease of AP from 0.2271 (without stemming) to 0.1716, after applying the light stemmer.

4.3. *N-grams & Trunc-n*

For all the models except DFR-I(n_e)C2 and *tf idf* the worst results are obtained by applying the 3-grams method. For DFR-I(n_e)C2 and *tf idf* models the worst performance is resulted from trunc-3 and then (with a small difference) by 3-grams. Even though *n*-grams seems to be an effective indexing strategy for languages such as Korean or Chinese (Abdou & Savoy, 2006), the results here show that, for Farsi, it is not more effective than word-based representations.

The bad performance of these methods when choosing a small value for *n* can be explained by the fact that truncating the words and leaving only the first three letters or splitting them into overlapping sequences of three characters causes lots of ambiguity. For example for Topic #75 ("مناطق دیدنی استان گلستان", "Tourist attractions in Golestan province") trunc-3 and 3-grams result too much ambiguity that the retrieval functions only based on the name of the province. As a result the second ranked retrieved document talks about meeting between the president and delegates of Golestan province. But by applying 6-grams or trunc-5 the terms "attraction" and "tourist" remain as their original form results more accurate matching.

From the results depicted in Table 2, it can be deducted that for *n*-grams and trunc-*n* approaches the bigger the value of *n* is the better is the performance. Actually as in Farsi words are usually not too long by choosing the bigger value for *n*, these approaches work like no stemming approach. In Table 3 the average of performance over the different 6 models shows that there is an increase of 7.8% when using trunc-5 compare to trunc-3 while 6-grams increase the average performance with 8.5% compared to 3-grams approach.

When looking at the results for all queries for DFR-I(n_e)C2 model it reveals that trunc-5 method gives a better AP than trunc-3 in 68 cases while 6-grams results a better AP than 3-grams in 54 cases. Table 4 presents, for 3 different queries, values of AP resulted from different values of *n*, in order to give an example of how AP increases by increasing the value of *n*. However this is not always the case as we can see in Table 5. One reason for this behaviour could be the existence of stems with 3 letters in the query so when using *n*-grams or trunc-*n* schemes with *n* equal to 3, it will be possible to obtain these stem with 3 letters from all the different forms in which they occurred in documents thus causing a better retrieval performance. For instance in Topic #565 ("خسارات خشک سالی" "Drought damages") there is the term "خشک سالی" "drought" which is a compound composed of two terms both having 3 letters, in Topic #571 ("اقتصاد جهانی نفت" "World fuel economy") the term "نفت" "Fuel" and in Topic #636 "آلودگی هوا" "Air pollution" the term "هوا" "Air" which again consists of 3 letters.

Table 4. AP for sample queries resulted from DFR-I(n_e)C2 model and *n*-grams and trunc-n strategies

	Average Precision						
	trunc-3	**trunc-4**	**trunc-5**	**3-grams**	**4-grams**	**5-grams**	**6-grams**
Topic #556	0.1213	0.1947	0.1944	0.0851	0.0952	0.1505	0.1953
Topic #595	0.2671	0.3446	0.3650	0.1844	0.3496	0.3597	0.3619
Topic #625	0.0978	0.1415	0.1458	0.0825	0.1085	0.1492	0.1540

Table 5. AP for sample queries resulted from DFR-I(n_e)C2 model and n-grams and trunc-n strategies

	Average Precision						
	trunc-3	**trunc-4**	**trunc-5**	**3-grams**	**4-grams**	**5-grams**	**6-grams**
Topic #565	0.6830	0.2565	0.1942	0.6797	0.5450	0.1745	0.1821
Topic #571	0.4240	0.4118	0.3506	0.4676	0.4435	0.4256	0.3463
Topic #636	0.2813	0.2310	0.2341	0.2500	0.2283	0.2269	0.2249

4.4. *Stoplist removal*

In order to analyse the effect of stoplist removal on retrieval effectiveness, the no stemmer strategy and the light stemmer were applied to the six IR models with and without applying stoplist removal. The results of these tests are shown in Table 6 and Table 7.

Table 6. MAP for no stemmer strategy with and without stoplist removal & change percentage for each model and for the average MAP

	Mean Average Precision						
	LM	**DFR-PL2**	***tf idf***	**DFR-I(n_e)C2**	**Okapi**	**Lnu-ltc**	**average**
no stem. /no stoplist	0.3449	0.3905	0.2156	0.4087	0.3815	0.3729	0.3524
no stemmer + stoplist	0.3592	0.4025	0.2648	0.4069	0.3962	0.3763	0.3677
Change %	+4.1%	+3.1%	+22.8%	-0.4%	+3.9%	+0.9%	+4.3%

Table 7. MAP for light stemmer strategy with and without stoplist removal & change percentage for each model and for the average MAP

	Mean Average Precision						
	LM	**DFR-PL2**	***tf idf***	**DFR-I(n_e)C2**	**Okapi**	**Lnu-ltc**	**average**
light stem. /no stoplist	0.3433	0.3982	0.2040	0.4117	0.3785	0.3737	0.3516
light stemmer + stoplist	0.3668	0.4155	0.2599	0.4168	0.4076	0.3874	0.3757
Change %	+6.8%	+4.3%	+27.4%	+1.2%	+7.7%	+3.7%	+6.9%

Obviously, stoplist removal helps to improve the retrieval effectiveness, as there is an increase of 4.3% (for no stemming) and 6.9% (for light stemmer) of average performance by applying stoplist removal. The results reveal that for both approaches without stoplist removal the DFR-I(n_e)C2 model has still the highest MAP compare to other IR models. In fact applying stoplist removal phase has a small impact on the MAP for this model (-0.4% for no stemming approach and 1.2%

for light stemmer). Performing the stoplist removal phase has its most impact on *tf idf* model. In this model as the term weighting depends on the term frequency, obviously in a text without noise a more accurate term weight and similarity calculation can be performed. For Okapi model (where there is also a remarkable change by adding stoplist removal) light stemmer with stoplist removal gives a better AP for 73 queries than without stoplist removal. For the same model and no stemming approach there is better AP results for 69 queries when applying stoplist removal compare to ignoring this step.

Taking the Okapi model and light stemmer as an example, when analysing each query separately it can be found Topic #646 (“اعزام به خارج از ایران” “Send off from Iran to abroad”) where the AP changed from 0.0010 (without stoplist removal) to 0.2389 (with stoplist removal), or Topic #638 (“موانع سرمایه گذاری در ایران” “Barriers to make an investment in Iran”) for which the performance changes from 0.0054 (without stoplist removal) to 0.1225 (with stoplist removal). On the other hand examples like Topic #609 (“بسته بندی میوه صادراتی” “Packing export fruit”) can be found where the AP is 0.2265 when stoplist removal is ignored compared to 0.0581 after stoplist removal. The reason of this behaviour for this particular topic is that the term “packing” in Farsi is a compound formed by two stems both included in the stoplist and thus being deleted from the topic after stoplist removal. This results in retrieval of documents about “export fruit” subject without covering “packing”. But such cases are not very often. In fact, cases where ignoring stoplist removal decreased the value of AP more than 10% is seen for only 3 queries.

Another issue that can be driven from the results in Table 6 and Table 7 is that applying only stoplist removal without performing any stemming approach gives a better result than applying the light stemmer without stoplist removal. The reason is again the many suffixes not attached to the words which are put in the stoplist and thus removed only by stoplist removal and not by the light stemmer.

5. Conclusion

From the obtained results in this experiment, having Farsi as the underlying language, the following conclusions can be drawn. In general IR models based on DFR paradigm are giving the best retrieval results for any stemming or indexing strategies. DFR-I(n_e)C2 was the best performing IR model followed by DFR-PL2. With the best performing model applying trunc-4, light stemmer and trunc-5, respectively, gives the best retrieving performance compare to other stemming or indexing approaches. Stemming approaches, either light stemmer or plural stemmer, improve the performance comparing to non-stemming approach with plural stemmer being less effective than the light stemmer. Trunc-*n* stemming strategy when *n* is equal to 4 or 5 also increases the retrieval performance compare to non-stemming approach. Putting above mentioned indexing and stemming strategies in increasing order of effectiveness the order would be, with a slight difference: trunc-5, light stemmer and trunc-4 and after all the plural stemmer. However, it can be deduced from the results that, for Persian language, either stemming or truncating does not make a significant improvement on performance. *N*-grams approach for any given *n* decreases the retrieval performance. The worst performance obtained by using 3-grams and trunc-3 approaches. For Persian language stoplist removal helps a lot to improve he retrieval performance. Actually for this language stoplist removal has a more positive impact on performance than applying light stemmer or the plural stemmer. The query-by-query analysis shows that during stemming phase there are some extreme situations happening because of some exceptions or rules in Persian morphology. These could be handled in several ways such as:

– Making a more precise morphological analysis in order to add extra rules to the stemming algorithm and hence enhancing the quality of suffix-removal process.

– Adding some extra controls on suffix-removal process such as defining names (personal, geographic, products, etc.) so as not to stem them.

– Adding certain rules in order to control the correctness of spelling after suffix removal (for cases where adding suffixes change the spelling).

– Taking account of part-of-speech (Savoy, 1993).

In addition some techniques of query expansion such as pseudo-relevance feedback (PRF or blind-query expansion) can be applied for this language to evaluate their effect on enhancing the retrieval effectiveness.

Acknowledgements

This work was supported in part by the Swiss NSF under Grant #200020-129535/1.

6. References

Abdou, S., & Savoy, J, “Statistical and comparative evaluation of various indexing and search models”, *AIRS*, vol. 4182, 2006, p. 362-373.

AleAhmad, A., Kamalloo, E., Zareh, A., Rahgozar, M., & Oroumchian, F., “Cross Language Experiments at Persian@CLEF 2008”, *LNCS*, vol. 5706, p. 105-112.

Amati, G., & Van Rijsbergen, C., “Probabilistic models of information retrieval based on measuring the divergence from randomness” *ACM Trans. Inf. Syst.* , vol. 20 no. 4, 2002, p. 357-389.

Bijankhan, M., Sheykhzadegan, J., Bahrani, M., & Ghayoomi, M., “Lessons from building a Persian written corpus: Peykar”, *Language Resources and Evaluation*, vol. 45 no. 2, 2011, p.143-164.

Dolamic, L., & Savoy, J., “Ad Hoc Retrieval with the Persian Language”, *CLEF 2009 proceedings*, 2009, p. 102-109

Dolamic, L., & Savoy, J., “Persian Language, Is Stemming Efficient?”, *DEXA 2009 proceedings*, 2009, p. 388-392.

Hiemstra, D., “Using language models for information retrieval”, CTIT Ph.D.Thesis, 2000.

McNamee, P., Mayfield, J., “Character n-gram tokenization for European language text retrieval”, *IR Journal*, vol. 7 no. 1-2, 2004, p. 73-97.

Manning, C. D., Raghavan, P., & Schütze, H., *Introduction to Information Retrieval*, Cambridge, Cambridge University Press, 2000.

Robertson, S. E., Walker, S., & Hancock-Beaulieu, M., “Experimentation as a way of life: Okapi at TREC”, *Inf. Process. Manage.* , vol. 36 no. 1, 2000, p. 95-108.

Savoy, J., “Stemming of French Words Based on Grammatical Categories”, *JASIS* , vol. 44 no. 1, 1993*,* p. 1-9.

Savoy, J., “A stemming procedure and stopword list for general French corpora”, *JASIS,* vol. 50 no.10, 1999, p. 944-952.

Savoy, J., “Combining multiple strategies for effective cross-language retrieval”, *IR Journal,* vol. 7 no. 1-2, 2004, p. 121-148.

Singhal, A., “AT & T at TREC-6”, *25th ACM Conference on Research and Development in Information Retrieval, ACM/SIGIR*, 2002, p. 35-41.