Pre-indexing Techniques in Arabic Information Retrieval

Souheila Ben Guirat[1,2,4], Ibrahim Bounhas[2,4] and Yahia Slimani[2,3,4]
[1]*Computer Sciences Department, Prince Sattam Bin Abdulaziz University, K.S.A.*
[2]*Laboratory of Computer Science for Industrial Systems, Carthage University, Tunisia*
[3]*Higher Institute of Multimedia Arts of Manouba (ISAMM), La Manouba University, Tunisia*
[4]*JARIR: Joint Group for Artificial Reasoning and Information Retrieval, Tunisia*

(www.jarir.tn)

Keywords: Arabic Information Retrieval, Hybrid Index, Statistical Modeling, Smoothing.

Abstract: Arabic document indexing is yet challenging given the morphological specificities of this language. Although there has been much effort in the field, developing more efficient indexing approaches is more and more demanding. One of the most important issues concerns the choice of the indexing units (e.g. stems, roots, lemmas, etc.) which both enhances retrieval efficiency and optimizes the indexing process. The question is how to process Arabic texts to retrieve the basic forms which better reflect the meaning of words and documents? In the literature several indexing units have been compared, while combining multiple indexes seems to be promising. In our previous works, we showed that hybrid indexes based on stems, patterns and roots enhances results. However, we need to find the optimal weight of each indexing unit. Therefore, this paper proposes to contribute in optimizing hybrid indexing. We compare and evaluate four pre-indexing methods.

1 INTRODUCTION

Indexing process aims to classify documents by content. Languages with sophisticated grammatical rules, such as Arabic, require sophisticated indexing methods.

Alhough there has been a great deal of Arabic document indexing, there are still indexing problems that have no been fully solved. One of he most imporatnt issues is to find the best index term that faithfully decribes the or user int original word.

2 RELATED WORKS

Identifying terms that discriminate and characterize the semantics of a document is the main goal of the statistical indexing (Andersson, 2003). In most related works in Arabic Information Retrieval (IR), documents are indexed using stems (Larkey and Connell, 2001; Aljlayl and Frieder, 2002; Chen and Gey, 2002) or roots (Al-Kabi et al., 2011; Al-Shawakfa et al., 2010; Khoja and Garside, 1999). While Arabic is characterized by its complex derivational and flectional morphology (Soudani et al., 2016, Wiem et al., 2015), literature surveys show that both indexing units may reach better results according to the experimental settings and the test collections (Elayeb and Bounhas, 2016). That is, combining several indexing units is promising and may reach better results (Ben Guirat et al., 2016). Consequently, the distinction between different indexing units (Hadni et al., 2012) is not an essential question. Anyhow, the most representative index types are combined in a hybrid indexing approach (Ben Guirat et al., 2016). However, we need to tune system parameters by assigning weights to different indexing units.

Various optimization techniques have been investigated for other languages including Chinese. As stated by Shi (2015) the problem of coping with term dependencies in Chinese is more pervasive than in most European languages where the bag-of-words approaches are still considered the state-of-the-art since they reached good results. In Chinese, however, phrases are not written as separated words but as continuous strings of characters. Shi et al. (2007) showed that combing unigrams with words and bigrams enhances Chinese IR. The proposed combining method was based on an empirical deter-

Ben Guirat, S., Bounhas, I. and Slimani, Y.
Pre-indexing Techniques in Arabic Information Retrieval.
DOI: 10.5220/0007393402370246
In *Proceedings of the 11th International Conference on Agents and Artificial Intelligence (ICAART 2019)*, pages 237-246
ISBN: 978-989-758-350-6

mination of the linear coefficients of each term.

Some works focused on post-combining approaches based on merging lists (Kwok, 1997; Leong and Zhou, 1997) or re-ranking and pseudo-relevance feedback (Luk and Wong, 2004; Yun et al., 2005). For example, Leong and Zhou (1997) and Kwok (1997) merged retrieval lists of words and bigrams to enhance search effectiveness. Tsang et al. (1999), Luk et al. (2001) and Chow et al. (2000) proposed a hybrid indexing approach based on bigrams and words. They assigned a weight equal to 1.5 for bigrams, while words are weighted according to their length.

To handle orthographic variants in Japanese, Kummer et al. (2005) combined words, N-grams, and Yomi-based indices across different document collections. From a computational point of view, they proposed a linear combination of the results of different retrieval systems and approaches. The contribution of each system is controlled by a weight. Relevance feedback is used to gradually optimize parameters, i.e. the weights of the individual indexes.

As far as Arabic is concerned, research in index combination is just starting. Ben Guirat et al. (2016) combined three indexing units, namely the root, the stem and the verbed-pattern. For example, the root of the word "الانقسامات" ("alinkissamat"; the divisions) is "ق س م" (k s m), its stem is "انقسام" (inkissam; division) and its verbed pattern is "انقسم" (inkassama; was divided). Our goal in this paper is to enhance hybrid indexing by adopting optimization in pre-indexing methods which have not been used in the field of Arabic IR.

3 PROPOSED WEIGHTING APPROACHES

As presented in the previous section, related works on Chinese and Japanese languages reveal the importance of combining more than one indexing unit. Besides, Ben Guirat et al. (2016) showed the effectiveness of hybrid indexing compared to single index-based Arabic IR.

That is, our goal is no longer showing the evidence of the importance of combining but finding the best weighting values for each of the indexing units.

In (Ben Guirat et al. 2016), post-indexing combining techniques were used. This slightly enhanced retrieval in Hybrid index IR compared to basic methods.

In the following, we describe pre-indexing combination approaches that we propose to further enhance Arabic IR. We mainly assess linear combination approaches and smoothing approaches.

3.1 Linear Combination Approaches

In this section, we try to aggregate the weights of stems (S_i), roots (R_i) and verbed patterns (P_i) to optimize the search process in the indexing phase. We propose to combine the frequencies of the three indexing units with a linear model, as follows:

$$I_j(W_i) = \alpha * TF_j(S_i) + \beta * TF_j(P_i) + \gamma * TF_j(R_i) \quad (1)$$

where $\alpha + \beta + \gamma = 1$.

$I_j(W_i)$ is the weight of the word Wi in the document dj and $TF_j(S_i)$, (respectively $TF_j(Pi)$ and $TF_j(R_i)$) is the normalized frequency of Si (respectively of Pi and Ri) in document dj.

α, β and γ are three parameters in the interval [0, 1], which may be varied or estimated in different manners.

The literature on optimization methods shows a variety of approaches. Some of the commonly used optimization approaches (Calandra et al., 2014) are compared in Table 1.

Gradient descent and Bayesian optimization reuire more computations which is not suitable for our combined indexing model that will be tested on a large amount of data.

Thus, from the list of methods in Table 1, we chose to implement grid and random search which seem to be more suitable to our problem because of their advantages and simplicity required in such hybrid IR system.

Grid search (Calandra et al., 2014) lies on running all the combinations of parameters (α and β) and computing the optimal value of a given IR metric. In this work, the chosen step size is 0.25. Using this step size, we aim to cover more values than previous combining work (Ben Guirat et al., 2016) which covered only 6 cases (compared to 12 cases in current work). Anyhow this will be refined in the random search method.

Grid search is costly given the high number of combinations. Random search tries to reduce the number of iterations. In this method, the set of samples is chosen randomly from all the possible combinations of discrete values of α, β and γ in [0, 1].

Table 1: Comparison of main optimization methods.

Method	Advantages	Drawbacks
Grid search	-Global optimum -Possible parallelization	-Combinatory -Grid refinement gives rise to new program iterations(gaps filling is not applicable)
Random search	-Global optimum -Possible grid adjustment -Less computational time -Possible parallelization -Rapid solution approximation	-Combinatory in case of global convergence
Bayesian optimization	-Global optimum -Probabilistic models allow to model noisy observations	-Combinatory
Gradient descent	-Faster convergence (First order optimizer)	-Requires additional computations to gradient evaluations and parameter initialization) -Local optimum. -Negative influence of parameter initialization in global convergence. -No possible gap filling

Our system implements the following algorithm.

```
j=1
While (j<threshold)
    Aj=RandomSampling (stepsize)
    If (performance (Aj)>performance
(Best))
        Best=Aj
    j++
End While
```

where Random Sampling (stepsize) is a random search variant of sampling. It generates a new position from the hypersphere of a given radius surrounding the current position.

The Random Search algorithm allows moving iteratively to better positions in the search space (Brownlee, 2011) .These positions are sampled from a hypersphere surrounding the current position. However, in this algorithm the step size significantly impacts results. To solve this problem, several random sampling approaches were proposed in literature (Brownlee, 2011).

In (Schumer and Steiglitz, 1968), Adaptive Step Size Random Search (ASSRS) reported the best results. It is a local search heuristic which changes dynamically the radius of the hypersphere around the best solutions to enhance accuracy and to avoid local optima (Gálvez et al., 2018). It attempts to heuristically adapt the hypersphere's radius: two new candidate solutions are generated, one with the current nominal step size and one with a larger step-size. The larger step size becomes the new nominal step size if and only if it leads to a larger improvement. If for several iterations neither of the steps leads to an improvement, the nominal step size is reduced.

In some recent studies, ASSRS is adopted because of its simplicity and high accuracy (Chen et al., 2015; Wessing et al., 2017; Gálvez et al., 2018). In our work, we initialize step size to 0.25 as in grid search. Then, we apply ASSRS to optimize this parameter and converge to the best configuration of indexing units weights.

3.2 Genetic Algorithm with Grid Search

As in (Cheung et al., 1997), we propose to combine genetic algorithm (Weise, 2009) with grid search for better performance and less calculations (Nyarko et al., 2014; Bergstra and Bengio, 2012).

In this method, we consider only two variables from (1). The proposed idea is based on problem composition by optimizing the value of α for each given value of β; then γ=1-α-β. This is implemented in the following algorithm.

```
Do
   Generate a set Sβ=( β1, β2, β3… βm)
    Generate a set Sα =(α1, α2, α3…αn)
    i=1
    While (i<n)
        j=1
        While (j<m)
                    Optimize (αi)
            j++
        End While
        i++
   End While
Until (No significant improvement is
observed)
```

3.3 Smoothing-based Combination

3.3.1 Smoothing Techniques

One of the possible ways to combine the indexing units is the smoothing technique. It refers to the adjustment of the maximum likelihood estimator of a language model so that it will be more accurate (Zhai and Lafferty, 2001).

Many smoothing algorithms have been proposed such as additive smoothing (Hazem and Morin, 2013), also called Laplace smoothing. It is one of the simplest smoothing types but its simplistic assumption model leads to many drawbacks including underestimating frequent n-grams and overestimating unseen ones (Hazem and Morin, 2013). Other alternatives are Good-Turing Estimator or Katz smoothing extending the intuition of Good-Turing. Jelineck-Mercer smoothing is also a well-known smoothing technique. These 4 previously named techniques all gave good results when tested for language n-gram modeling (Hazem and Morin, 2013).

Another empirical comparison of smoothing techniques in language modeling (Chen and Goodman, 1996), considering multiple set sizes, performed multiple runs for both bigram and trigram models. Its results proved again that Katz and Jelineck-Mercer smoothing perform consistently well. Church-gale smoothing, which combines Good Turing with bucketing, outperforms them with bigrams.

The same work proposes two novel methods: average count (an instance of Jelinek-Mercer) and one count method (combining to intuition Makay and Petro (Chen and Goodman, 1996)). Despite of the bad performance of one count method, it gives better results than the other methods.

Besides, Zhai and Lafferty (2001) compared Jelineck-Mercer, Bayesian smoothing using Dirichlet Priors (Laplace is a special case for this technique) and absolute discounting (based on the similar idea as Jenileck-Mercer). This comparison aimed to find out the best technique for language models applied to Ad hoc IR and showed that Dirichlet Priors is desirable for estimation issues while Jenileck-Mercer idea suits more query modeling.

Based on (Federico et al., 2008; Koehn, 2009), Witten Bell smoothing (Bell et al., 1990) is considered as well established smoothing technique as it out-performs many smoothing techniques.

However, many comparison works (Chen and Goodman, 1996; Federico et al., 2008; Koehn, 2009) showed that improved Kneser-Ney gives always the best results an used to perform well even in the interpolated Kneser-Ney (The, 2006).

3.3.2 Index Weighting as a Smoothing Problem

We inspired this model from interpolated Kneser–Ney smoothing. We consider a word represented by a triplet (S_i, P_i, R_i) as a trigram. Our goal is to compute the weight of each stem S_i based on its frequency and the frequencies of its verbed pattern and its root. We have:

$$\mathrm{SM_j(w_i)} = \mathrm{SM_j(S_i, P_i, R_i)}$$

$$= \frac{\mathrm{c'_j(S_i)}}{\mathrm{c_j(S_i)}} + \mathrm{D} * \left[\frac{\mathrm{c'_j(P_i)}}{\mathrm{c_j(P_i)}} + \mathrm{D} * \left(\frac{\mathrm{c'_j(R_i)}}{\mathrm{c_j(R_i)}} + \mathrm{D}\right)\right] \quad (2)$$

$\mathrm{c'_j(S_i)}$ is given by:

$$\mathrm{c'_j(S_i)} = \max\left(0, \mathrm{c_j(S_i)} - \mathrm{D}\right) \quad (3)$$

In the same manner, we compute $c'(R_i)$and $c'(P_i)$.

If $\mathrm{c_j(S_i)} = 0$, we consider that $\frac{\mathrm{c'_j(S_i)}}{\mathrm{c_j(S_i)}}$=0. This applies also for $\frac{\mathrm{c'_j(P_i)}}{\mathrm{c_j(P_i)}}$ and $\frac{\mathrm{c'_j(R_i)}}{\mathrm{c_j(R_i)}}$.

D is an absolute constant in the interval [0, 1], for which we may experiment different values. By default (Stolcke and al., 2011), it is computed as follows:

$$\mathrm{D} = \frac{\mathrm{n_1}}{\mathrm{n_1} + 2 * \mathrm{n_2}} \quad (4)$$

where n_1(respectively n_2) is the total number of triplets (in our case this is equivalent to the number of stems) with have exactly one (respectively two) occurrences in d_j.

The weight of the stem is obtained by normalizing $SM_j(w_i)$:

$$\mathrm{I_j(w_i)} = \frac{\mathrm{SM_j(w_i)}}{\sum_k \mathrm{SM_j(w_k)}} \quad (5)$$

4 EXPERIMENTS

4.1 Test Collection

We tested our approaches in the LDC's standard test collection ("Arabic Newswire Part 1", catalog number LDC2001T55). It is composed of 869 megabytes of news articles taken from "Agence France Presse" (AFP) Arabic newswire i.e. 383,872 articles dated from May 13, 1994 through December 20, 2000.

Two versions of TREC topics were developed in 2001 (25 topics) and 2002 (50 topics). Each topic contains 3 parts, namely a title, a description and narrative. The later contains further description that may help the human analyst.

As in some previous works (Ben Guirat et al., 2016), authors used a modified version of Ghwanmeh stemmer (Gwanmeh et al., 2009), that appeared to be more efficient than other stemmers in comparative studies (Al-Shawakfa et al., 2010). It achieves better results compared to Khoja and Larkey stemmers (Ben Guirat et al., 2016). We use PL2 (Poisson estimation for randomness), which is implemented in Terrier platform (Ounis et al., 2006), as ranking model.

4.2 Evaluation Protocol

Referred to previous works on LDC2001T55 collection (Soudi et al., 2007), we assess two scenarios i.e. using only titles and combining titles with descriptions. We perform four experimental setups as detailed in table 2.

For each experimental setup, we perform six runs (cf. Table 3). For measuring search effectiveness, our comparison is based on 4 metrics, namely Recall, Precision at 10, Mean Average Precision (MAP) and R-Precision (Zingla et al., 2018).

Table 2: Experimental setups.

Designation	TREC version	# topics	Query type
T1	TREC 2001	25	Title
T2	TREC 2002	50	
TD1	TREC 2001	25	Title + description
TD2	TREC 2002	50	

Precision is equal to the fraction of documents retrieved that are relevant to the query, while recall is the percentage of relevant documents that are successfully retrieved. To study the ability of our system to rank documents, we evaluate recall and/or precision at several positions. For example, Precision at 10 stands for precision computed for the 10 top ranked documents. R-precision is equal to precision at *R* which is equal to the number of relevant documents for a given query. In the same perspective, average precision (AVP) allows to evaluate system performance by considering precision and recall at every position in the ranked list. MAP is the average value of AVP computed for all queries.

4.3 Experimentations Results

Table 3: Compared methods.

Approach	Method	Label
Baselines	Stem based-indexing	S1
	Pattern based-indexing	S2
	Root based-indexing	S3
Hybrid indexing	Grid Search based combination	H1
	Random Search based combination	H2
	Genetic-based combination	H3
	Kneser-Ney Smoothing	H4

In the following, we start by parameter tuning in grid search and Kneser-Ney methods (cf. section 1). Then we compare our approaches using standard IR metrics.

4.3.1 Grid Search and Kneser-Ney Parameters Tuning

The goal of this step is to find out the best parameter configuration that grid search and Kneser-Ney smoothing technique may reach. We compare the MAP values of the different values of parameters (cf. Table 4; Table 5).

Table 4: MAP values in grid search.

Parameter values			T1	T2	TD1	TD2
α	β	γ				
0.00	0.25	0.75	0.0136	0.0137	0.0126	0.0120
0.00	0.5	0.50	0.0138	0.0127	0.0130	0.0114
0.00	0.75	0.25	0.0129	0.0152	0.0152	0.0126
0.25	0.00	0.75	0.1601	0.1975	0.1634	0.2355
0.25	0.25	0.50	0.2230	0.2235	0.2214	0.2597
0.25	0.50	0.25	0.2556	0.2269	0.2558	0.2700
0.25	0.75	0.00	0.2421	0.1994	0.2446	0.2477
0.50	0.00	0.50	0.2613	0.2538	0.2654	0.2927
0.50	0.25	0.25	0.2995	0.2728	0.3080	0.3114
0.50	0.50	0.00	0.2924	0.2682	0.3037	0.3079
0.75	0.00	0.25	0.2895	0.2683	0.2966	0.3107
0.75	0.25	0.00	0.3055	0.2764	0.3164	0.3161

Table 4 shows that the worst MAP values in grid search are obtained by the first configurations, especially when the stem weights are null and give better results when the stem weights increase. This fact is due to the nature of the different indexing units; thus giving important weights to stems which are naturally the most canonical forms of words, yields to better MAP values.

Table 4 also shows that all test setups reach the best results for the same configuration noted GS12 (α=0.75, β=0.25, γ=0). This configuration will be used in the remaining comparative studies.

Table 5: MAP values variation with *D* parameter tuning.

	MAP			
D parameter	T1	T2	TD1	TD2
Default	0.0047	0.0293	0.0023	0.0044
0,1	0.0006	0.0006	0.0004	0.0005
0,2	0.0004	0.0077	0.0003	0.0004
0,3	0.0077	0.0167	0.0002	0.0002
0,4	0.0655	0.0416	0.0001	0.0002
0,5	0.1023	0.1373	0.0670	0.092
0,6	0.0442	0,1056	0.0243	0.0462
0,7	0.0136	0.0648	0.0057	0.0193
0,8	0.005	0.0374	0.0024	0.0093
0,9	0.003	0.024	0.0016	0.056
1	0.0023	0.0171	0.0013	0.0034

In Table 5, we tested the *D* parameter values from 0 to 1 using a step= 0.1 but also the default value (see Eq. 4). The MAP values of all these possible showed that D=0.5 gives the best MAP value. So it will be further compared to other combining techniques in next parts.

4.3.2 MAP Values

In this section, we study the MAP value which is one of the most important criteria used in IR systems performance comparison. H1 usually gives high MAP values. Moreover, it gives the best in T1 and TD1 setups (cf. Figures 2 and 4). However, S2 gives the best results in 2002 queries. It may be explained by the number and the length of the queries as well as the pool size variations between the two TREC versions (Voorhees, 2002). Indeed, the average pool size in 2001 was 164.9 and did not exceed 118.2 in 2002. Kneser-Ney smoothing did not fit our combining goal and usually gives the worst MAP values.

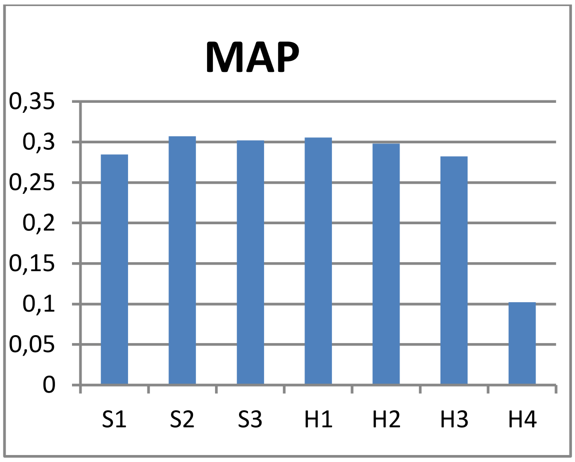

Figure 1: MAP values (T1: LDC 2001 titles).

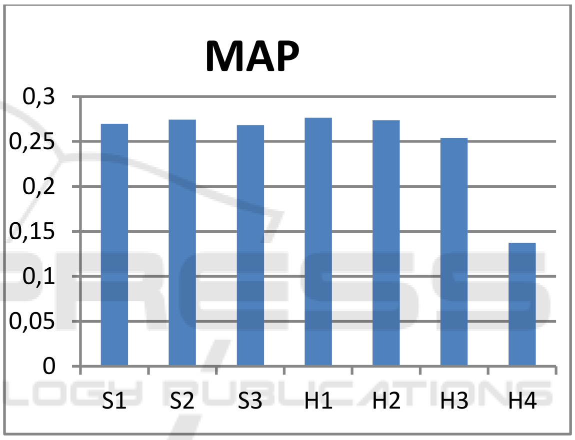

Figure 2: MAP values (T2: LDC 2002 titles).

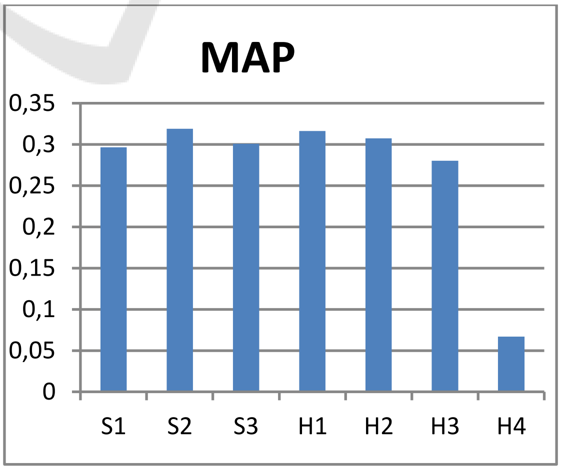

Figure 3: MAP values (TD1: LDC 2001 titles + Descriptions).

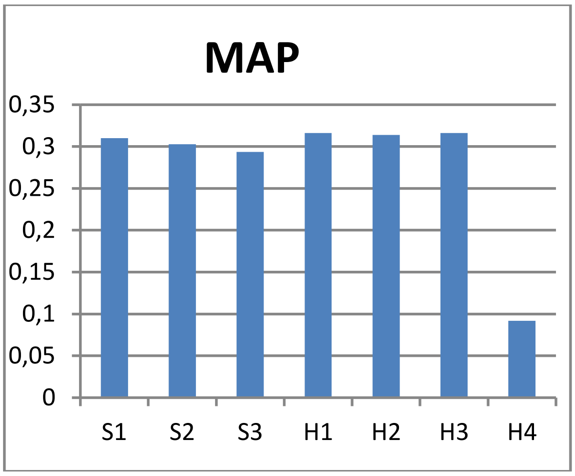

Figure 4: MAP values (TD2: LDC 2002 titles + Descriptions).

4.3.3 Precision at 10 Values

In this section, we study the values of precision at 10. Figures 5 and 6 show that root and pattern-based methods have the worst precision rates compared to the hybrid methods. Furthermore, H3 usually gives good result that overcomes for all approaches in T1 and TD2.

Besides, all the hybrid approaches (except Kneser-Ney smoothing method) generally give better P10 values compared to the baselines except in T2 (cf. Figure 6) where S1 also gives comparable MAP value.

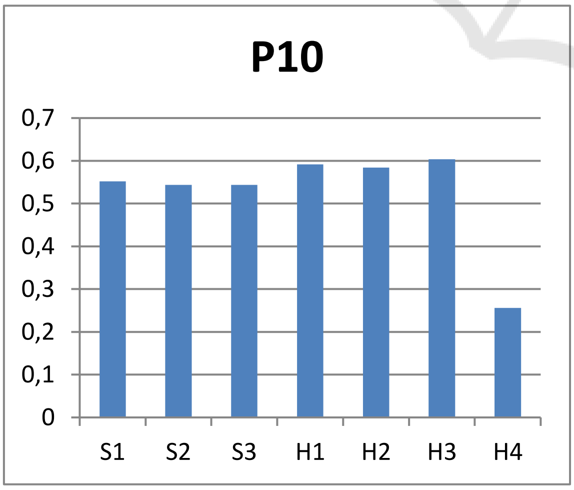

Figure 5: Precision at 10 values (T1: LDC 2001 titles).

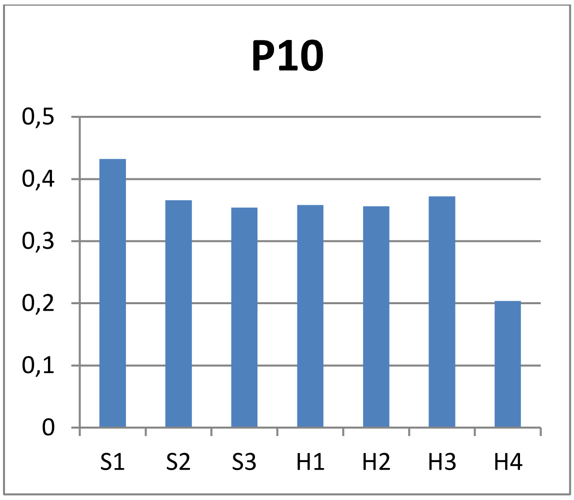

Figure 6: Precision at 10 values (T2: LDC 2002 titles).

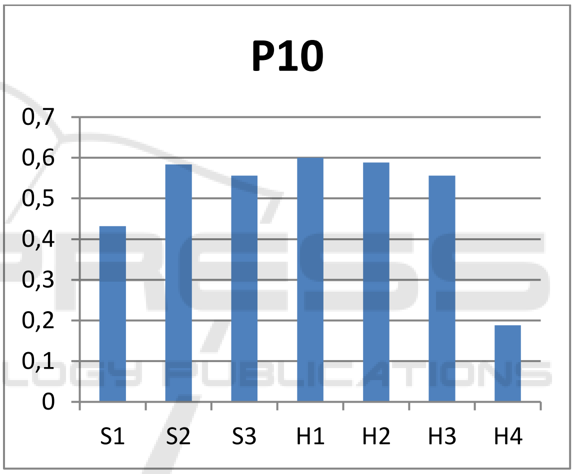

Figure 7: Precision at 10 values (TD1: LDC 2001 titles + Descriptions).

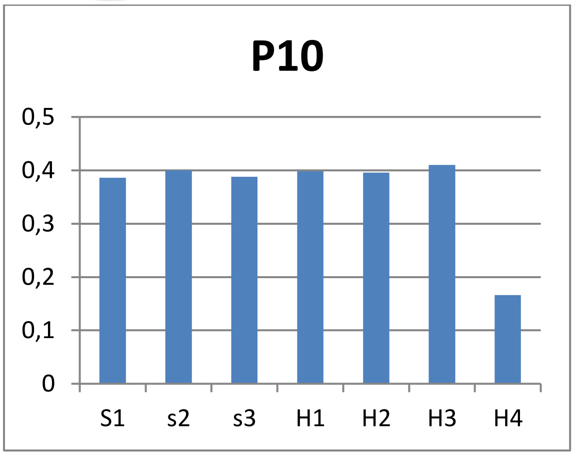

Figure 8: Precision at 10 values (TD2: LDC 2002 titles + Descriptions).

4.3.4 R-Precision and Recall Values

Table 6: R-precision and recall Results comparison.

Setup	Approach	R-Precision	Recall
T1	S1	0.3271	0.9471
	S2	0.3401	0.9609
	S3	0.3325	0.9730
	H1	0.3401	0.9602
	H2	0.3398	0.9570
	H3	0.3212	0.9667
	H4	0.1457	0.9228
T2	S1	0.2943	0.9756
	S2	0.2955	0.9788
	S3	0.2957	0.9801
	H1	0.3008	0.9788
	H2	0.3017	0.9773
	H3	0.2799	0.9801
	H4	0.1766	0.9768
TD1	S1	0.3393	0.9876
	S2	0.3534	0.9917
	S3	0.3443	0.9958
	H1	0.3507	0.9917
	H2	0.3460	0.9900
	H3	0.3269	0.9956
	H4	0.0933	0.9454
TD2	S1	0.3372	0.9961
	S2	0.3215	0.9971
	S3	0.3151	0.9977
	H1	0.3390	0.9969
	H2	0.3396	0.9962
	H3	0.3161	0.9976
	H4	0.1362	0.9874

Table 6 compares simple and hybrid approaches based on two main criteria, namely the R-precision and recall in all the test setups. For simple indexing methods, we naturally notice that root-based-indexing (S3) always reaches the best recall results.

Moreover, we notice the improvement given by the hybrid approaches compared to basic methods. Thus, hybrid indexing always reaches better R-Precision, except in TD1 when S2 gives better R-Precision results while the chosen smoothing technique did not improve the IR performance.

Furthermore, using descriptions in queries usually enhances R-Precision and recall compared to title-based queries. Actually, descriptions enlarge the scope of the query by additional terms which may be synonyms or variants of those existing in the title.

Finally, focusing on the different results of 2001 and 2002 test setups, we note that 2002 queries give always better results than 2001. This may be explained by the improvement of number of runs that have been submitted to TREC 2002 (Soudi, 2007) which enhanced relevance judgment and the quality of the final collection.

5 CONCLUSION

The contribution of this paper is to use optimization and smoothing techniques in order to assign weights to system parameters in the pre-indexing stage.

To get a closer representation of the importance of each indexing unit in representing word meaning, we used the LDC's standard test collection which is covering more vocabulary than ZAD collection used in previous combining works (Ben Guirat et al., 2016). This test collection also contains more concise queries, which include detailed descriptions that gave us the opportunity to study the effect of query length in retrieval effectiveness. This allowed us to obtain better results and study the specificities of each approach/configuration.

The presented results clearly show that our proposed approaches which combine different indexing units usually outperform simple indexing. Especially, the grid search usually gives the best performance with its optimal weights values (α=0.75, β=0.25, γ=0). However, the variety of the number of queries and their length shows variations between the 4 setup results.

Further, we would like to assess other combining methods like other smoothing techniques (Zhai and Lafferty, 2001), since Kneser-Ney smoothing did not improve our IR system. Regression approaches (Lamprier et al., 2007) could also be used to estimate the units' weights values. We also plan to integrate other stemming tools to process texts. For instance, stem-based indexing with FARASA (El Mahdaouy et al., 2018) and lemma-based indexing with MADAMIRA enhanced Arabic IR (Soudani et al., 2018).

REFERENCES

Abdelhadi Soudi, Antal van den Bosch, and Gunter Neumann, 2007. "*Arabic Computational Morphology: Knowledge-based and Empirical Methods*", Springer Publishing Company, Incorporated.

Abdelkader El Mahdaouy, Said El Alaoui Ouatik, and Éric Gaussier, 2018. "*Improving Arabic information retrieval using word embedding similarities*". I. J. Speech Technology 21(1): 121-136.

Aitao Chen, and Frederic Gey, 2002. "Building an Arabic stemmer for information retrieval", *Proceedings of the Text Retrieval Conference TREC-11, pp. 631-639.*

Akemi Gálvez, Andres Iglesias, Iztok Fister, Iztok Fister Jr., Eneko Osaba, and Javier Del Ser, 2018. "*Memetic Modified Cuckoo Search Algorithm with ASSRS for the SSCF Problem in Self-Similar Fractal Image Reconstruction*". *HAIS 2018: 658-670.*

Amir Hazem and Emmanuel Morin, 2013. “A Comparison of Smoothing Techniques for Bilingual Lexicon Extraction from Comparable Corpora”, *Proceedings of the Workshop on Building and Using Comparable Corpora (BUCC’13).*

Andreas Stolcke, Jing Zheng, Wen Wang and Victor Abrash, 2011. “SRILM at sixteen: Update and outlook”. *In Proceedings IEEE Automatic Speech Recognition and Understanding Workshop*.

Bernard K. S. Cheung, André Langevin, and Hugues Delmaire, 1997. “*Coupling genetic algorithm with a grid search method to solve mixed integer nonlinear programming problems*”, Computers and Mathematics with Applications. vol. 34, no. 12,pp. 13-23.

Bilel Elayeb, and Ibrahim Bounhas, 2016. “*Arabic Cross-Language Information Retrieval: A Review*”. ACM Trans. Asian & Low-Resource Lang. Inf. Process. 15(3): 18:1-18:44.

Chengxiang Zhai and John Lafferty, 2001. “A study of smoothing methods for language models applied to Ad Hoc information retrieval”, *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval, p.334-342, New Orleans, Louisiana, USA*.

Ellen M. Voorhees, 2002. “Overview of TREC 2002”. *In Proceedings of the 11th Text Retrieval Conference (TREC 2002)*, NIST Special Publication 500-251, National Institute of Standards and Technology.

Emad Al-Shawakfa, Amer Al-Badarneh, Safwan Shatnawi, Khaleel Al-Rabab’ah, and Basel Bani-Ismail, 2010. “*A comparison study of some Arabic root finding algorithms*”, Journal of the American Society for Information Science and Technology, vol. 6, no. 5, pp. 1015–1024..

Emmanuel K. Nyarko, Robert. Cupec, and Damir Filko, 2014. “*A Comparison of Several Heuristic Algorithms for Solving High Dimensional Optimization Problems*”, International journal of electrical and computer engineering systems, Vol.5. No.1, pp. 1-8.

Iadh Ounis, Gianni Amati, Vassilis Plachouras, Ben He, Craig Macdonald, and Christina Lioma, 2006. “Terrier: A High Performance and Scalable Information Retrieval Platform”. *In Proceedings of Open Source Information Retrieval (OSIR) Workshop, Seattle, USA.*

James Bergstra, and Yoshua Bengio, 2012. “*Random search for hyper-parameter optimization*”, The Journal of Machine Learning Research, vol. 13, no. 1, pp.281-305.

Jason Brownlee, 2011. “*Clever Algorithms: Nature-inspired Programming Recipes*”, Lulu.com.

Ken C.W.Chow, Robert W. P Luk, Luk and K.am-Fai Wong, Kuilam Kwok, 2000. “Hybrid term indexing for different IR models”, *Proceedings of the 5th International Workshop Information Retrieval with Asian Languages, pp.49-54*.

Kuilam Kwok, “Comparing Representations in Chinese Information Retrieval”, 1997. *Proceedings of 20th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 34-41, Philadelphia, PA, USA .*

Leah Larkey, and Margaret E. Connell, 2001. “Arabic information retrieval at UMass in TREC-10”, *Proceedings of Text Retrieval conference (TREC) 2001, Gaithersburg, USA.*

Linda Andersson, 2003. “*Performance of Two Statistical Indexing Methods, with and without Compound-word Analysis”*, English summary of a bachelor thesis, Stockholm University, Sweden, available online:www.nada.kth.se/kurser/kth/2D1418/uppsatser 03/LindaAndersson_compound.pdf.

Lixin Shi, Jian Y. Nie, Jing Bai, 2007. “Comparing Different Units for Query Translation in Chinese Cross-Language Information Retrieval”, *Proceedings of the 2nd International Conference on Scalable Information Systems (Infoscale), Suzhou, China.*

Lixin Shi, 2015. “*Relating dependent terms in information retrieval*”, Ph.D thesis, Monreal University, Canada.

Ma Schumer, and Kenneth Steiglitz, 1968. “Adaptive step size random search”. *IEEE Transactions on Automatic Control, 13(3), 270-276.*

Marcello Federico, Nicola Bertoldi and Mauro Cettolo, 2008. “IRSTLM: An open source toolkit for handling large scale language models”, *INTERSPEECH 2008, 9th Annual Conference of the International Speech Communication Association, Brisbane, Australia.*

Mariem Amina Zingla, Chiraz Latiri, Philippe Mulhe, Catherine Berrut, and Yahia Slimani, 2018. “*Hybrid query expansion model for text and microblog information retrieval*”, Information Retrieval Journal, pp 1–31, Inf Retrieval J, Springer Netherlands.

Meryeme Hadni, Abdelmonaime Lachkar, and Said O. Alaoui, 2012. “A New and Efficient Stemming Technique for Arabic Text Categorization”, *ICMCS 2012 International Conference on Multimedia Computing and Systems, pp. 791 - 796, Tangier, Morocco*.

Mohamed Aljlayl and Ophir Frieder, 2002. “On arabic search: improving the retrieval effectiveness via a light stemming approach”, *the 11th Conference on Information and Knowledge Management, McLean, Virginia, USA, pp. 340-347.*

Mohammed Al-Kabi, Qasem Al-Radaideh and Khalid Akawi, 2011. “*Benchmarking and assessing the performance of Arabic Stemmers*”, Journal of Information Science, vol. 37, no. 2, pp. 1–12.

M-K. Leong, and H. Zhou, 1997. “Preliminary qualitative analysis of segmented vs bigram indexing in Chinese”, *Proceedings of the Sixth Text Retrieval Conference (TREC-6), Gaithersburg, Maryland, USA.*

Nadia Soudani, Ibrahim Bounhas, and Sawssen Ben Babis, 2018. “Ambiguity Aware Arabic Document Indexing and Query Expansion: A Morphological Knowledge Learning-Based Approach*”. FLAIRS Conference 2018: 230-23.*

Nadia Soudani, Ibrahim Bounhas,and Yahia Slimani, 2016. “Semantic Information Retrieval: A Comparative Experimental Study of NLP Tools and

Language Resources for Arabic". *ICTAI 2016: 879-887*.

Nina Kummer, Christa Womser-Hacker, Noriko Kando, 2005. "Handling Orthographic Varieties in Japanese Information Retrieval: Fusion of Word-, N-gram, and Yomi-Based Indices across Different Document Collections*", Proceedings of the Second Asia conference on Asia Information Retrieval Technology, pp. 666-672, Jeju Island, Korea*.

Philipp Koehn, "*Statistical Machine Translation*", Cambridge University Press; 1 edition, 17 December 2009.

Robert W. P. Luk, Kam-Fai Wong, and Kuilam Kwok, 2001. "Hybrid term indexing: an evaluation", *Proceedings of the 2nd NTCIR Workshop on Research in Chinese & Japanese Text Retrieval and Text Summarization, pp.130-136, Tokyo Japan.*

Robert W.P. Luk and Kam-Fai Wong, 2004. "Pseudo Relevance Feedback and Title Re-ranking for Chinese Information Retrieval", *Proceedings of the Fourth NTCIR Workshop on Research in Information Access Technologies, Information Retrieval, Question Answering and Summarization, Tokyo, Japan.*

Roberto Calandra, Nakul Gopalan, André Seyfarth, Jan Peters, and Mark P. Deisenroth, 2014. "Bayesian gait optimization for bipedal locomotion", in " Learning and Intelligent Optimization", *proceedings of 8th International Conference, Lion 8, Gainesville, FL, USA, February 16-21, 2014 (Revised Selected Papers) Lecture Notes in Computer Science, pp. 274-290.*

Sameh Ghwanmeh, Saif Addeen Rabab'ah, Riyad Al-Shalabi, and Ghassen Kanaan, 2009. "Enhanced Algorithm for Extracting the Root of Arabic Words". *Sixth International Conference on Computer Graphics, Imaging and Visualization, pp. 388–391, Tianjin, China.*.

Shereen Khoja, and Roger Garside, 1999. "*Stemming Arabic Text. Technical report",* Computing department, Lancaster University, U.K.

Simon Wessing, Rosa Pink, Kai Brandenbusch, and Gunter Rudolph, 2017. "Toward Step-Size Adaptation in Evolutionary Multiobjective Optimization". *EMO 2017: 670-684*.

Souheila Ben Guirat, Ibrahim Bounhas, and Yahia Slimani, 2016. *"Combining Indexing Units for Arabic Information Retrieval"*. International Journal of Software Innovation (IJSI), vol. 4, no. 4, pp. 1-14.

Souheila Ben Guirat, Ibrahim Bounhas, and Yahia Slimani, 2016. "A Hybrid Model for Arabic Document Indexing", *in 17th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD 2016), May 30 - June 1, 2016, Shanghai China*.

Souheila Ben Guirat, Ibrahim Bounhas, and Yahia Slimani, 2018. "Enhancing Hybrid Indexing for Arabic Information Retrieval", *32nd International Symposium, ISCIS 2018, Held at the 24th IFIP World Computer Congress, WCC 2018, Poznan, Poland*.

Stanley F. Chen and Joshua Goodman, 1996. "An empirical study of smoothing techniques for language modeling*", Proceedings of the 34th annual meeting on Association for Computational Linguistics, p.310-318, Santa Cruz, California*.

Sylvain Lamprier, Tassadit Amghar, B. Levrat, and Frederic Saubion, 2007. "Document Length Normalization by Statistical Regression", *19th IEEE International Conference on Tools with Artificial Intelligence(ICTAI 2007), Patras, Greece*.

T.F. Tsang, R.W.P. Luk, and K.F. Wong, 1999. "Hybrid term indexing using words and bigrams", *Proceedings of the fourth International Workshop on Informatoon Retrieval with Asian Languages (IRAL), pp. , 112-117, Academia Sinica, Taiwan.*

Thomas Weise, 2009. *"Global Optimization Algorithms - Theory and Application"*, Self-Published, second edition.

Timothy C. Bell, John G. Cleary and Ian H. Witten, 1990. *"Text Compression*". Prentice Hall, Englewood Cliffs, N.J,

Wiem Lahbib, Ibrahim Bounhas, and Yahia Slimani. 2015. "Arabic Terminology Extraction and Enrichment Based on Domain-Specific Text Mining". *ICTAI 2015: 340-347.*

Yee Whye The, 2006. "*A Bayesian interpretation of interpolated Kneser-Ney*", Technical Report TRA2/06, School of Computing, National University of Singapore.

Yu Chen, Jian Cao, and Pinglei Guo, 2015. "CPU Load Prediction Based on a Multidimensional Spatial Voting Model". *DSDIS 2015:* 97-102.

Yun Xiao, Robert W.P. Luk, Kam-Fai. Wong, Kuilam Kwok, 2005. "Some Experiments with Blind Feedback and Re-ranking for Chinese Information Retrieval", *Proceedings of the Fifth NTCIR Workshop Meeting on Evaluation of Information Access Technologies: Information Retrieval, Question Answering and Cross-Lingual Information Access, Tokyo, Japan.*