Report on the TREC-11 Experiment: Arabic, Named Page and Topic Distillation Searches

Jacques Savoy, Yves Rasolofo

Institut interfacultaire d'informatique, Université de Neuchâtel (Switzerland)
E-mail: {Jacques.Savoy, Yves.Rasolofo}@unine.ch

Summary

This year we took part in the Arabic cross-language information retrieval track (for us limited to monolingual Arabic retrieval) and also in both named page and topic distillation searches. In the last two tasks, we made use of link anchor information and document content in order to construct Web page representatives. This document representation uses multi-vectors in order to highlight the importance of both link anchor information and document content.

Introduction

Today the IR community is faced with a new paradigm and many exciting challenges with regards to Web page searches. Some of these include: managing huge volumes of documents via distributed IR models, crawling across the Web in order to find appropriate Web sites to index, accessing documents written in various languages, measuring the quality or authority of available information, providing answers to user requests that are often very short and expressed in ambiguous terms, satisfying a large range of search types (ad hoc, question-answering, location of online services, topic distillation, known-item and interactive searches for specific document types, or satisfying specific geographical or time constraints).

In this context, the first part of this paper presents our monolingual Arabic retrieval model. Section 2 describes our procedures for indexing and retrieving Web pages based on two document representations, and our distributed indexing framework based on the Okapi probabilistic model. Section 3 explains the IR approach we use when combining both Web page content and anchor information when searching for specific named pages. Finally, Section 4 describes how our IR scheme can be used within the context of topic distillation task.

In order to evaluate our hypothesis, we used the SMART system as a testbed, implementing various vector-space IR schemes and the Okapi probabilistic model (Robertson et al., 2000). This year our experiments were conducted on an Intel Pentium III/600 (memory: 1 GB, swap: 2 GB, disk: 6 x 35 GB) and all experiments were fully automated.

1. Arabic Information Retrieval

During the last two CLEF evaluation campaigns we suggested various IR models and tools for handling different European languages (Savoy, 2002a; 2002b). This year we expanded upon our knowledge by adding the Semitic language family, which includes Arabic.

The IR model we are proposing for Arabic text searches involves indexing both documents and queries, based on the words described in Section 1.1, or using n-gram segmentation, as presented in Section 1.2. Section 1.3 shows how we can combine result lists provided using various indexing and searching schemes that process the same document collection. The last section provides an account of the retrieval effectiveness achieved by various IR models and also that of various combined approaches. The diverse IR tools may be seen on our Web site (www.unine.ch/info/clef/).

1.1. Word-Based Indexing

In order to effectively search Arabic documents, we first convert and normalize the Arabic Unicode characters into Latin letters (a technique used in Malta where the Arabic language is written using the Latin alphabet). Due to variations in morphological rules or geographical traditions however many Arabic characters have more than one Unicode representation. For example there are different forms of alef (alef madda "آ", alef hamza above "أ", alef hamza below "إ"), transliterated within our approach into the same letter "A" (see Table 1). Our conversion procedure is based on but is not identical to that used in previous work by Darwish et al. (2002). There are of course some questionable assignments such as the hamza letter "ء", which is considered equivalent to the alef maksura "ى". In this phase some Unicode characters have also been removed (e.g., diacritic marks that are usually optional in newspapers such as tatweel "ـ", fathatan "ً" or various punctuation marks "؟". These diacritic marks may however be important in other contexts in order to resolve underlying ambiguity (e.g., in legal documentation)).

In a second step we ignore words appearing in our Arabic stopword and Arabic stoplist (the latter contains 347 words, available at www.unine.ch/info/clef/).

In a third step we automatically remove both prefixes and suffixes to form our Arabic stems. These relatively light stemming approaches are similar to those suggested by Larkey & Connell (2002) or Lackey et al. (2002). As shown in Table 2, our stemmer "stem2" is more conservative while our "stem3" represents a more aggressive affix-stripping process. The word length must be greater than a given threshold (between 4 and 6 letters) before we will remove a given affix. Some of our rules are questionable and our stemmers must be viewed, as they are only a first draft.

Arabic	Latin	Arabic	Latin
ء ى	Y	إ أ آ ا	A alif
ؤ و	w waw	ئ ي	y yeh
ب	b beh	ة	p teh
ت	t teh	ث	v theh
ج	j jeem	ح	H hah
خ	x khah	د	d dal
ذ	O thal	ر	r reh
س	s seen	ش	P sheen
ص	S sad	ض	D dad
ط	T tah	ظ	Z zah
ع	E ain	غ	g ghain
ف	f feh	ق	q qaf
ك	k kaf	ل	l lam
م	m meem	ن	n noon
ه	h heh	ي	X yeh
ز	z zain		

Table 1. Our Arabic letter transliterations

	Remove from front	Remove from rear
stem2	fAl, XAl, bAl, wAl, Al, w, Y	An, At, hA, Yp, Yh, Yn, wn, Y, p, h
stem3	wAl, fAl, bAl, Al, ll, bt, yt, lt, mt, tt, wt, st, nt bm, lm, wm, km, fm, wA, fA, lA, bA, wy, ly, sy, fy, t, y, m, b, n, l, k, w, A, f	km, tm, At, An, wn, wh, hn, hm, wA, tA, hA, nA, tk, ty, th yn, yh, yp p, h, y, t, k, A

Table 2. Main rules used by our two Arabic stemmers

1.2. N-gram Indexing

As an alternative procedure we can index Arabic documents using 3-gram (or tri-gram) indexing (Darwish & Oard, 2002). In this case, each word is replaced by a set of three-letter sequences. For example the word "document" will be replaced by {"doc", "ocu", "cum", "ume", "men" and "ent"}. In our current implementation we do not stem words before splitting them into tri-grams and we also remove very frequent tri-grams (obtained from our stopword list).

1.3. Data Fusion

We use a single search model (or engine) when searching document collections. We might however suggest sending the request to different search engines that handle the same document collection but that use different indexing or search schemes. Finally, once we have obtained result lists from these various search engines, we need to merge them in an effective manner (data fusion problem). Thus, even though certain degrees of retrieval effectiveness may be attributed to each search approach, combining the result lists might provide better average precision. If we were to use RSV_k to denote the retrieval status value (or document score) for a given document retrieved by the k*th* search engine, Fox & Shaw (1994) suggest using various operators (see Table 3) and show that the best performance can be achieved using "combSUM".

combMAX	MAX (RSV_k)
combMIN	MIN (RSV_k)
combSUM	SUM (RSV_k)
combANZ	SUM (RSV_k) / # of nonzero (RSV_k)
combNBZ	SUM (RSV_k) * (# of nonzero (RSV_k))
combRSV%	SUM (RSV_k / maxRSV)
combRSVn	SUM[(RSV_k-minRSV)/(maxRSV-minRSV)]

Table 3. Data fusion strategies

Of course we might also employ the round-robin merging strategy whereby we take the first retrieved item from the first result list, then the first retrieved document from the second list, etc., and finally the first item from the last result list and then back again to the first results list, thus providing the next item to be put in the final list. Duplicates encountered in this process are simply ignored.

1.4. Evaluation

We evaluated various vector-space schemes (see Appendix 1 for detailed specifications of these models) together with the Okapi probabilistic model. As shown in Table 4a, we also evaluated our two light stemmers and the tri-gram indexing scheme using short queries ("Title" or T), medium-size queries (constructed using the Title and Descriptive logical sections or TD) or long queries (based on Title, Descriptive and Narrative sections or TDN).

An examination of this data shows that the best average precision is obtained using the Okapi model while second best results are usually obtained using the vector-space model "Lnu-ltc", and the "dtu-dtn" scheme usually ranks third. Moreover, it seems that our "stem2" stemmer performs slightly better than the more aggressive "stem3" procedure (the mean difference over 10 retrieval models was 4.7% for T queries, 6.1% for TD queries, and 7.4% for TDN queries). On average, the tri-gram indexing scheme seems to be a little bit less effective than word-based indexing (using the "stem3" or "stem2" stemming approaches). Note however that with the Okapi model, the tri-gram approach performed better for shorter queries (Title only) or TD requests.

From previous evaluations on different European languages (Savoy, 2002a), it is clearly apparent that requests containing more search terms provide improved average precision (from "Title" to TD, with a mean improvement of around 13.3% and a mean difference between "Title" and TDN of around 17.5%). With the Arabic corpus these differences appear to fall within a similar range, as shown in Table 4a. For example, using the Title only evaluation as a baseline, performance can be improved by about 9.9% for TD queries (mean over 10 retrieval models, using stem2) or 4.9% with TD requests (mean over 10 retrieval models, using tri-grams). When comparing short request query performances (Title only and TDN), the mean difference over 10 IR models is around 18.6% (stem2) or 15.6% (stem3). For tri-gram models however the average precision differences are around -2.1%, due to the poor performance by the "nnn-nnn" and "bnn-bnn" approaches during TDN queries.

	Average Precision								
	Title			TD			TDN		
Model	Word stem2	Word stem3	3-grams no stem	Word stem2	Word stem3	3-grams no stem	Word stem2	Word stem3	3-grams no stem
Okapi-npn	**27.41**	**26.09**	**28.77**	**30.51**	**29.22**	**31.45**	**32.87**	30.40	**29.95**
Lnu-ltc	25.80	24.93	25.95	29.88	28.68	29.79	32.33	**30.41**	29.49
dtu-dtn	24.92	24.35	23.98	27.25	25.22	27.96	28.45	27.30	26.30
atn-ntc	22.71	21.30	22.76	24.44	22.42	24.00	25.98	25.47	22.44
ltn-ntc	24.19	22.94	22.65	26.45	24.77	24.50	27.94	26.60	21.58
lnc-ltc	20.66	19.55	20.46	24.77	23.38	25.10	29.58	27.05	27.39
ntc-ntc	20.27	19.09	19.64	23.03	21.46	23.92	25.38	23.39	23.26
ltc-ltc	18.41	17.90	19.73	21.33	20.30	23.95	26.25	24.43	25.40
nnn-nnn	12.65	12.11	8.22	13.84	13.21	6.31	14.84	13.60	5.10
bnn-bnn	12.47	11.64	11.29	10.86	9.92	5.90	8.54	7.00	1.62

Table 4a. Average precision of various IR models using the Arabic corpus (monolingual)

	Average Precision								
	Title			TD			TDN		
#doc/#term	Word stem2	Word stem3	3-grams no stem	Word stem2	Word stem3	3-grams no stem	Word stem2	Word stem3	3-grams no stem
Okapi-npn	27.41	26.09	28.77	30.51	29.22	31.45	32.87	30.40	29.95
d=5 / t=10	31.51	31.13	32.30	33.62	32.46	33.56	35.21	33.06	32.66
d=5 / t=20	32.45	31.91	32.40	34.15	33.31	34.60	35.75	33.59	33.67
d=5 / t=30	32.98	31.85	32.49	34.23	33.24	35.30	36.25	33.86	33.96
d=5 / t=40	33.34	31.75	33.37	34.47	33.27	35.68	36.36	34.07	34.19
d=5 / t=50	33.26	31.64	33.35	34.45	33.20	35.78	36.47	34.12	34.07
d=5 /t=100	33.32	31.54	32.62	34.64	33.19	36.19	36.40	**34.27**	**34.34**
d=5 /t=150	33.07	31.32	32.23	34.48	33.22	**36.24**	36.39	33.87	34.17
d=10 /t=10	32.39	31.57	32.75	34.03	32.81	33.52	35.27	32.68	32.37
d=10 /t=20	33.35	32.41	34.44	34.78	33.53	34.72	36.01	33.69	32.69
d=10 /t=30	33.82	32.78	**34.78**	35.23	33.88	35.04	36.39	33.60	32.85
d=10 /t=40	34.13	**32.97**	34.71	35.52	**34.08**	35.39	36.46	33.49	33.03
d=10 /t=50	**34.20**	32.89	34.54	35.66	34.08	35.50	36.42	33.40	33.14
d=10/t=100	34.00	32.17	34.56	**35.85**	33.69	35.70	36.52	33.56	33.42
d=10/t=150	33.75	31.76	34.35	35.46	33.40	35.36	**36.55**	33.41	33.41
d=25 /t=10	32.06	31.33	31.91	33.75	32.21	32.72	34.99	32.53	31.61
d=25 /t=25	33.09	32.81	32.02	34.59	33.03	33.66	35.74	33.04	32.07
d=25 /t=50	33.78	32.96	32.55	35.24	33.21	34.01	36.01	33.54	32.47
d=25/t=100	33.88	32.90	32.57	35.42	33.09	34.11	36.07	33.45	32.52

Table 4b. Average precision using blind query expansion (Okapi model, Arabic corpus, monolingual)

Model	Average Precision								
	Title no expand	TD no expand	TDN no expand	Title + expand 1 / 1 / 1	TD + expand 1 / 1 / 1	TDN + expand 1 / 1 / 1	Title + expand 1 / 1 / 1.5	TD + expand 1 / 1 / 1.5	TDN + expand 1 / 1 / 1.5
Best single	28.77	31.45	32.87	34.78	36.24	36.55	34.78	36.24	36.55
Round-robin	28.68	31.40	32.28	35.21	36.81	36.59	35.21	36.81	36.59
combRSVn	30.19	33.13	**33.60**	**36.54**	**36.90**	36.94	**36.75**	**37.16**	**36.98**
combMAX	**30.79**	**33.58**	31.78	35.25	36.56	**37.12**	34.44	36.57	34.85
combSUM	29.69	33.03	33.23	35.98	36.65	36.63	36.31	37.01	36.59
combRSV%	29.02	32.63	32.85	36.00	36.33	36.27	36.04	36.61	36.09
combNBZ	29.07	32.78	33.02	35.94	36.58	36.36	36.08	37.05	36.48
combANZ	27.26	28.34	27.67	33.83	32.78	33.67	27.17	22.54	23.88
combMIN	20.52	20.77	20.09	30.29	27.14	27.85	18.41	11.94	14.26

Table 5. Average precision of various data merging strategies (Arabic corpus, monolingual)

Run name	Query	Okapi stem3	Okapi stem2	Okapi 3-grams	Average precision
UniNE1	T-D	doc=10 / term=15	doc=10 / term=75	doc=10 / term=20	**37.12**
UniNE2	T	doc=10 / term=40	doc=5 / term=20	doc=25 / term=15	35.72
UniNE3	T-D-N	doc=5 / term=50	doc=10 / term=40	doc=10 / term=20	38.07
UniNE4	T-D	doc=10 / term=15	doc=10 / term=75	doc=10 / term=20	36.60

Table 6. Specifications and evaluation of our official monolingual Arabic runs

Pseudo-relevance feedback (or blind-query expansion) has proven to be a useful technique for enhancing retrieval effectiveness. In this study, we adopted Rocchio's approach (Buckley et al., 1996) with $\alpha = 0.75$, $\beta = 0.75$ whereby the system was allowed to add t terms extracted from the k best ranked documents obtained from the original query. We used the Okapi probabilistic model to evaluate this proposal and then enlarged the query by 10 to 75 terms taken from the 5 or 10 best-ranked articles (see Table 4b). From examining the data in this table we were able to conclude that overall blind-query expansion does indeed improve retrieval performance. For example, with short requests the improvement is +21.5% when applying the stem2 stemmer, +23.1% with stem3 and +16% with the tri-gram model.

Finally, we evaluated various data fusion strategies that might be employed to improve retrieval effectiveness. In our case we used the same document collection and simply submitted the same request to our three search engines (stem2, stem3, and tri-grams). Based on the data in Table 5, it appears that data fusion based on combRSVn performs better when blind query expansion is taken into account. The combMAX strategy seems to be the more appropriate solution when we ignore pseudo-relevance feedback. However, both combRSVn and combSUM seem to be more robust data fusion operators. We should however mention that the percentage improvement over the best single approach is not really significant (e.g., in Title only queries without query expansion, the combRSVn increases the average precision by +4.9% compared to +2.2% when using TDN requests). Finally, the last three columns of Table 5 show the results of the same three individual runs where we instead multiplied each document score obtained from the tri-gram model by 1.5, without modifying document scores for both the word-based indexing schemes.

Table 6 lists the exact specifications for our official runs. These runs were carried out using different numbers of documents and terms during blind query expansions but all runs were built using the combRSVn operator and multiplying the tri-gram document scores by 1.5.

2. Our Okapi Search Model

As shown in the preceding section, the Okapi search model provides significantly greater retrieval effectiveness. However, in order to manage the Web collection (1,247,753 documents as extracted from the .GOV domain, or about 18.1 GB of data), we needed to modify this search model for two reasons. First, we wanted to incorporate two document representatives for each Web page, and secondly we needed to distribute the inverted file in order to respect the 2 GB limit.

When using multiple document representations, the retrieval status value (or document score denoted $RSV(D_i)$) was calculated as follows (inner product):

$$RSV(D_i) = \sum_{j=1}^{m} w_{ij} \quad qw_j$$

within which w_{ij} indicates the weight assigned to the term T_j in the document D_i, qw_j the indexing weight assigned to the same term in the current query and m the number of search keywords.

When processing two (or more) document representations, we estimated the degree of similarity between the document D_i and the current query as a linear combination of the inner product of the two document representations to be given as:

$$RSV(D_i) = \alpha \sum_{j=1}^{m} w_{ij}^{(1)} \, qw_j + (1-\alpha) \sum_{j=1}^{m} w_{ij}^{(2)} \, qw_j \quad (1)$$

where $w^{(1)}_{ij}$ indicates the weight attached to the term T_j in the document D_i in the first document representation (and $w^{(2)}_{ij}$ for the second document surrogate), and α a parameter used to assign a comparative importance to the first document representative as relative to the second.

Thus, by assigning α a value close to 1.0, we give more importance to the first document representation. At the limit, setting $\alpha = 1.0$ implies that we ignore the second document surrogate.

Creating a single inverted file from a collection of around 18 GB might be impossible using a 32-bit system (e.g., Linux). To overcome this limit, we will concentrate on the last scheme, and in this case we will follow the approach described in (Rasolofo & Savoy, 2003), whereby we merge the result lists obtained from searching different collections (collection fusion problem). This is achieved by using the document scores computed by each collection as a key for the merging and sorting process.

3. Named Page Searching

When submitting a request to a search engine, sometimes users do not want a ranked list of Web pages regarding a particular topic but rather they would prefer the location of an underlying service or known-item (usually presented in a short list of the most probable locations). For example, an appropriate answer to the requests "US passport renewal", "Maryland unemployment insurance benefits" or "FBI's most wanted list" does not consist of a ranked list of documents about these subjects but rather the site(s) containing the required form/information/list. To accomplish this goal we need to implement an IR system that can retrieve a short number of pages (at the limit, only one) corresponding to the user's request. In this context of known-item search, the underlying IR system must clearly place the emphasis on precision.

3.1. Search Models

As a basis for our search model we used the Okapi model as described in Section 2. Our first document representative was based on information found in the Web page and its corresponding <TITLE> and <META> tags ("keywords" and "description"). Web pages might of course also contain links and their anchor texts (or anchor texts for outgoing links) and this combined set of internal textual information would thus form the first representatives of these pages.

On the other hand, previous studies (Craswell et al., 2001), (Westerveld et al., 2002), (Kraaij et al., 2002) have shown that anchor texts from other Web pages pointing to the current page provide compact and often accurate descriptions of the current page's content. Thus link anchor texts extracted from all Web pages pointing to the current page were concatenated to form our second document representative. To this second surrogate we also added the text contained in the current page's <TITLE> tag (with the text delimited by this tag appearing in both document representatives). Finally, we might also consider the URL content (or more precisely, the similarity between the URL text and the request, or also the URL length). This additional source of information has not taken into account in our current search models.

3.2. Evaluation

In this IR search model based on two document representatives, we first needed to determine the relative importance assigned to the first document representative (based on internal Web page content) as compared to the weight attached to the second document surrogate (based mainly on link anchor texts from Web pages pointing to the current one). This relative importance for each surrogate is controlled by the α parameter (see Section 2). When $\alpha = 1.0$, we would only account for internal textual representation while when setting $\alpha = 0.5$, we would attribute equal importance to both document representatives.

Number of queries	150
Number of relevant doc.	170
Mean rel. doc. / request	1.133
Standard deviation	0.378
Median	1
Maximum	3 (q#: 9, q#: 145)
Minimum	1

Table 7. Relevance judgment statistics (named page searches, TREC-11)

Our evaluation will be based on the mean reciprocal rank (MRR) of the first correct answer found by the system. Table 7 depicts statistics on the relevance assessments of this test-collection, clearly showing that we usually obtain one correct answer per topic. For each of the 150 queries we considered only the first 100 retrieved items. As seen in Table 8, the best α value

seems to be around 0.6, thus assigning a little more weight to internal representation.

Run name	MRR	# in top 10	# not found
α = 0.0	0.5046	99 (66.0%)	27 (18.0%)
α = 0.1	0.5507	106 (70.67%)	24 (18.0%)
α = 0.2	0.5929	109 (72.67%)	22 (14.67%)
α = 0.3	0.6366	115 (76.67%)	16 (10.67%)
α = 0.4	0.6491	120 (80.0%)	12 (8.0%)
α = 0.5	0.6688	120 (80.0%)	7 (4.67%)
α = 0.6	**0.6800**	122 (81.33%)	**7** (4.67%)
α = 0.7	0.6775	**125** (83.33%)	7 (4.67%)
α = 0.8	0.6772	124 (82.67%)	9 (6.0%)
α = 0.9	0.6511	121 (80.67%)	11 (7.33%)
α = 1.0	0.5867	116 (77.33%)	16 (10.67%)

Table 8. IR model evaluation for various combinations of two document representatives (no stemming)

Run name	MRR	# in top 10	# not found
α = 0.0	0.4991	99 (66.0%)	25 (16.67%)
α = 0.3	0.6163	118 (78.67%)	15 (10.0%)
α = 0.5	0.6506	120 (80.0%)	9 (6.0%)
α = 0.7	0.6735	123 (82.0%)	6 (4.0%)
α = 0.8	0.6710	122 (81.33%)	6 (4.0%)
α = 1.0	0.5771	116 (77.33%)	13 (8.67%)

Table 9. IR model evaluation for various combinations of our two document representatives (S-stemming)

When high precision results are required for indexing documents or requests, it is usually not a good idea to include a stemming procedure. However a very light stemming procedure might only be adopted when removing the final "s" in English (such stemming is called "S-stemming" (Harman, 1991)). Thus the words "house" and "houses" will be reduced to the same root while the term "housing" will be treated as a different indexing unit.

A comparison of results depicted in Table 9 (S-stemming) to those in Table 8 (no stemming) indicates that performance differences are rather small. Better performances can however always only achieved from IR approaches that ignore the stemming phase.

Finally, Table 10 provides a summary description of our four official runs. Only the UniNEnp3 run needed an additional comment. This run was based on UniNEnp1 and after obtaining a ranked list, we reranked the first ten retrieved items according to the number of matches between the query terms and the corresponding Web page's title field (however, such a strategy does not improve retrieval effectiveness).

Run name	MRR	Description
UniNEnp1	**0.636**	No stemming, α = 0.3
UniNEnp2	0.616	S-stemming, α = 0.3
UniNEnp3	0.625	Reranking the first 10 items
UniNEnp4	0.504	No stemming, α = 0.0

Table 10. Description of official named-page runs

4. Topic Distillation Searches

Under the label "topic distillation", we had to implement an IR scheme able to find a list of key resources on a given topic. Explicitly defining what does or does not constitute a good key resource is however difficult, and each definition seems to become more ambiguous. Of course, Web pages with appropriate content might be considered as good key resources and we could retrieve them using a classical IR model. On the other hand, key resources may also be good hubs (or Web pages pointing to different pages containing pertinent content with respect to the submitted request). Moreover, if a Web page is linked to two, three or more sons having a high degree of similarity with the request, it seems more appropriate to return this father page rather than the two, three of more sons. More generally however returning many pages extracted from the same Web site would not be viewed as a wise strategy. Thus to suggest a proper solution for this specific task, we decided to employ different strategies capable of pointing to reliable starting points for browsing rather than simply retrieving Web pages with good content. An overview of such strategies that might be applied in a Web environment can be found in Savoy & Rasolofo (2001).

4.1. Search Models

As for the task of named page searching, we built two document representatives for each Web page contained in the .GOV collection. The first representative accounted for Web page content along with its <TITLE> and <META> tags ("keywords" and "description") plus all link anchor texts extracted from other pages pointing to this current page. The second document representative was built from the text delimited by the <TITLE> tag together with link anchor texts from all outgoing links. These two document representations may be useful both for accounting for the content of both the Web page (first surrogate) and other pages accessible within a one-click distance from the current page (our second representative).

Once the pages are retrieved, we followed hyperlinks coming into them in order to define proper starting points for browsing (in this case we followed existing hyperlinks in the reverse orientation). To retrieve these

starting points we used our spreading activation (SA) searching scheme (Savoy, 1996), (Crestani & Lee, 2000), (Savoy & Picard, 2001). Using this method, document scores initially computed by the IR system (denoted RSV(D_i)), are propagated to the linked documents through a certain number of cycles, using a propagation factor. We used a simplified version with only one cycle and a fixed propagation factor λ for all links. Thus the final retrieval status value for a document D_i linked to k documents is computed using the following equation:

$$RSV'(D_i) = RSV(D_i) + \lambda \cdot \sum_{j=1}^{k} RSV(D_j) \qquad (3)$$

When trying in our experiments to extract the proper starting sites for browsing, we only considered all incoming links for each of the k best-ranked documents (in this paper the constant k was fixed to 200 and the parameter λ to 0.35).

As an alternative, we assumed that the first k top-ranked items would form a "root set" or a kernel of pertinent pages from which we could consider all incoming and all outgoing links in order to form an extended set (called the base set) of pages that might be of interest for a given topic. Based on Kleinberg's HITS algorithm, we assumed that a Web page pointing to many other information sources must be viewed as a "good" hub while a document with many Web pages pointing to it must be viewed as a "good" authority. Likewise, a document that points to many "good" authorities is an even better hub while a Web page pointed to by many "good" hubs is an even better authority (Kleinberg, 1998).

For document D_i after c+1 iterations, the updated formulas for the hub and authority scores $H^{c+1}(D_i)$ and $A^{c+1}(D_i)$ are:

$$A^{c+1}(D_i) = \sum_{D_j=parent(D_i)} H^{c}(D_j)$$

$$H^{c+1}(D_i) = \sum_{D_j=child(D_i)} A^{c}(D_j)$$

which is computed for the k best-ranked documents (defined as the root set) retrieved by a classical search model, together with their children and parents (which defined the base set). The hub and authority scores were updated for five iterations (while the ranking did not change after this point), and a normalization procedure (dividing each score by the sum of all square values) was applied after each step.

As other possibilities, we might consider the PageRank algorithm (Brin & Page, 1998) or probabilistic argumentation systems (Picard, 1998).

4.2. Evaluation

In order to evaluate the performance of a topic distillation IR scheme, we could use the precision achieved after retrieving 5 or 10 documents (under the labels "Prec@5" or "Prec@10") together with the number of relevant items retrieved (out of a total of 1,574 for the 49 queries included in the .GOV collection).

Number of queries	49
Number of relevant doc.	1,574
Mean rel. doc. / request	32.122
Standard deviation	37.33
Median	22
Maximum	188 (q#: 558)
Minimum	1 (q#: 588)
Number of distinct roots / query	
Mean	9.429
Standard deviation	15.27
Median	13
Maximum	64 (q#: 596)
Minimum	1 (q#: 581)
URL length 1	31
length 2	194
length 3	536
length 4	402
length 5	263
length 6	110
length 7 and more	38
# pertinent items file	1,380
# pertinent items path	194

Table 11. Relevance judgment statistics (topic distillation searching task, TREC-11)

Table 11 shows various statistics based on relevance assessments. The mean number of relevant items (or key resources) per request is 32.122. From considering the number of distinct roots (e.g., the first part of an URL, e.g., "trec.nist.gov"), we find that in mean, there were 9.4 different roots per query (for Query# 581, all relevant items coming from the root page "www.cancer.gov"). On the other hand, for Query# 558, we found 26 relevant pages extracted from the root page "www.whitehouse.gov" (and of these, 25 were from "www.whitehouse.gov/news/releases/2001/").

In our first set of experiments, we evaluated our extended Okapi IR model (see Section 2). By varying the value attached to the α parameter, we assigned more or less weight to each document representation. More precisely, when we set $\alpha = 0.0$, we accounted for text delimited by the <TITLE> tag and all link anchor texts from outgoing links. In other words, we viewed the page as a good starting point for browsing (limited however to one-click distance). On the other hand,

when α = 1.0, our search model was based on Web page content and from the various link anchor texts contained in all pages pointing to this particular document.

Table 12a displays the various results produced by our IR model (without stemming) when varying the relative importance of each document representative. From this data, the best α value seemed to be around 0.9, based upon the precision achieved after 10 retrieved items (or 0.7 for 5 retrieved records). Thus, our first representation (content-oriented) seems to be more valuable for this specific IR task. Data in Table 11 seems to confirm these findings, given the various statistics on relevance assessments used in this task. For example, of the 1,574 pertinent items, 1,380 (or 87.7%) correspond to a filename while only 194 (or 12.3%) to subdirectories or path entries (URLs ending with a "/" or with "index.htm" or similar terms). Moreover, URLs of unitary length (or roots) correspond to only 31 (or 2%) relevant items.

Run name	Prec@5	Prec@10	rel. & retr.
α = 0.0	15.92	13.06	457
α = 0.1	17.14	14.69	562
α = 0.2	18.37	15.92	626
α = 0.3	18.78	17.35	683
α = 0.4	20.82	17.14	793
α = 0.5	21.22	17.96	926
α = 0.6	22.04	19.39	982
α = 0.7	**24.08**	19.59	973
α = 0.8	22.86	21.43	**991**
α = 0.9	22.86	**21.63**	965
α = 1.0	23.67	18.37	919

Table 12a. Evaluation of various document representatives combinations (no stemming, TD queries)

Run name	Prec@5	Prec@10	rel. & retr.
α = 0.0	11.84	9.39	635
α = 0.2	18.37	13.47	877
α = 0.4	21.63	16.12	1,217
α = 0.6	20.82	18.16	**1,231**
α = 0.8	**22.04**	**18.98**	1,159
α = 1.0	22.04	17.35	1,064

Table 12b. Evaluation of various document representatives combinations (no stemming, TDN queries)

When we considered longer queries (built using the Title, Descriptive and Narrative logical sections), retrieval performance seemed to decrease relative to the precision achieved upon retrieving 5 or 10 items. Of course this value clearly increases for longer requests, as shown by the number of relevant and retrieved records (last column of Table 12b).

Our UniNEdi1 run is based on short requests (Title only) while our UniNEdi3 run is based on the same processing but for TDN queries. For both runs, after retrieving content-based Web pages using our extended Okapi model, we applied spreading activation with λ = 0.35 for the first k = 200 top-ranked items. Following this stage, we pruned the retrieved URL (keeping only three URLs per site).

Using the SA method and based on the best run data shown in Table 12a, we tried various parameter settings as depicted in Table 13. Clearly, the propagation factor λ must be smaller than 0.35, and the SA must be limited to the first 50 best-ranked items (instead of k = 200).

Parameters	Prec@5	Prec@10	rel. & retr.
no stem, α = 0.9	22.86	21.63	965
λ = 0.01, k = 50	23.27	21.43	1,020
λ = 0.025, k = 50	**25.71**	**21.84**	**1,020**
λ = 0.05, k = 50	25.31	21.63	1,020
λ = 0.1, k = 50	22.86	19.39	1,020
λ = 0.15, k = 50	20.82	18.37	1,020
λ = 0.2, k = 50	19.59	16.73	1,020
λ = 0.1, k = 25	22.86	19.39	1,000
λ = 0.1, k = 75	22.04	18.37	1,029
λ = 0.1, k = 100	20.00	18.16	1,036
λ = 0.1, k = 200	15.10	15.71	1,051

Table 13. Evaluation of various parameter settings for the spreading activation approach

For the UniNEdi2 run, we applied the Kleinberg's HITS algorithm in order to define hub and authority pages (k = 200), and to form our ranked list we summed the hub and authority scores of each Web page, defining the new document score. Finally we pruned the retrieved URL.

Using the best run from Table 12a as the starting point, we varied the number k of the top-ranked items included in the root set from the HITS method, as shown in Table 14. The data in this table seems to clearly indicate that in this task the HITS algorithm does not perform well, whatever the value of k, whether we account for the hub score, the authority score or both.

Finally, Table 15 provides a summary description of our five official runs, all of which were created without a stemming procedure. Searching for good browsing starting points when using the SA or Kleinberg approaches clearly fails, or more precisely searching key resource does not means searching for browsing proper starting points.

Parameters	Prec@5	Prec@10	rel. & retr.
no stem, α = 0.9	22.86	21.63	965
k = 50, hub score	3.67	3.27	526
k = 50, auth. score	6.12	4.90	526
k = 50, both	2.86	3.27	526
k = 100, hub score	2.86	2.45	684
k = 100, auth. score	**5.31**	**3.88**	679
k = 100, both	2.45	2.45	683
k = 150, hub score	2.04	1.84	762
k = 150, auth. score	4.90	3.88	742
k = 150, both	2.04	2.04	765
k = 200, hub score	1.22	1.22	771
k = 200, auth. score	4.49	2.86	717
k = 200, both	1.22	1.22	730
k = 300, hub score	0.41	0.82	643
k = 300, auth. score	3.67	2.45	576
k = 300, both	0.82	0.83	598

Table 14. Evaluation of different parameter settings for the HITS algorithm

Run name	Prec@10	description
UniNEdi1	8.37	UniNEdi5 + SA (λ=0.35)
UniNEdi2	3.27	UniNEdi5 + HITS
UniNEdi3	7.76	TDN, no stem, α = 0.7, SA
UniNEdi4	14.29	UniNEdi5 + reranking
UniNEdi5	19.59	no stemming, α = 0.7

Table 15. Description of our official named page runs

Only the UniNEdi4 run needs any additional comments. This run is based on UniNEdi5 and after we obtained a ranked list, we computed and sorted the Web sites according to number of pages present in the top 50 best-ranked items. Following this step, we selected pages from those sites having the greatest number of matches between the query terms and the underlying URL texts (however, this selection and reranking procedure did not improve the retrieval effectiveness).

Acknowledgments

The authors would like to thank S. Abdou for his help in understanding some aspects of the Arabic language and C. Buckley from SabIR for allowing us the opportunity to use the SMART system. This research was supported by the SNSF (Swiss National Science Foundation) under grants 21-58'813.99 and 21-66'742.01.

References

Brin, S. & Page, L. (1998). The anatomy of a large-scale hypertextual Web search engine. Proceedings WWW8, 107-117.

Buckley, C., Singhal, A., Mitra, M. & Salton, G. (1996). New retrieval approaches using SMART. Proceedings TREC-4, NIST Publication #500-236, 25-48.

Craswell, N., Hawking, D. & Robertson, S. (2001). Effective site finding using link anchor information. Proceedings ACM-SIGIR'2001, 250-257.

Crestani, F. & Lee, P.L. (2000). Searching the Web by constrained spreading activation. *Information Processing & Management*, 36(4), 585-605.

Darwish, K. & Oard, D.W. (2002). Term selection for searching printed Arabic. Proceedings ACM-SIGIR'2002, 261-268.

Darwish, K., Doermann, D., Jones, R., Oard, D.W. & Rautiainen, M. (2002). TREC-10 experiments at Maryland: CLIR and video. Proceedings TREC-10, NIST Publication #500-250, 549-561.

Fox, E.A. & Shaw, J.A. (1994). Combination of multiple searches. Proceedings TREC-2, NIST Publication #500-215, 243-249.

Harman, D. (1991). How effective is suffixing? *Journal of the American Society for Information Science,* 42(1), 7-15.

Kleinberg, J. (1998). Authoritative sources in a hyperlinked environment. Proceedings ACM-SIAM Symposium on Discrete Algorithms, 668-677.

Kraaij, W., Westerveld, T. & Hiemstra, D. (2002). The importance of prior probabilities for entry page search. Proceedings ACM-SIGIR'2002, 27-34.

Larkey, L.S., Ballesteros, L. & Connell, M.E. (2002). Improving stemming for Arabic information retrieval: Light stemming and co-occurrence analysis. Proceedings ACM-SIGIR'2002, 275-282.

Larkey, L.S. & Connell, M.E. (2002). Arabic information retrieval at UMass in TREC-10. Proceedings TREC-10, NIST Publication #500-250, 562-570.

Picard, J. (1998). Modeling and combining evidence provided by document relationships using PAS systems. Proceedings ACM-SIGIR'1998, 182-189.

Rasolofo, Y. & Savoy, J. (2003). Term proximity scoring for keyword-based retrieval systems. Proceedings ECIR-03, to appear.

Robertson, S.E., Walker, S. & Beaulieu, M. (2000). Experimentation as a way of life: Okapi at TREC. *Information Processing & Management*, 36(1), 95-108.

Savoy, J. (1996). Citation schemes in hypertext information retrieval. In *Information retrieval and hypertext*, M. Agosti, A. Smeaton (Eds), Kluwer, 99-120.

Savoy, J. & Picard, J. (2001). Retrieval effectiveness on the Web. *Information Processing & Management*, 37(4), 543-569.

Savoy, J. & Rasolofo, Y. (2001). Report on the TREC-9 experiment: Link-based retrieval and distributed collections. Proceedings TREC-9, NIST Publication #500-249, 579-588.

Savoy, J. (2002a). Report on CLEF-2001 experiments: Effective combined query-translation approach. In C. Peters, M. Brachler, J. Gonzalo, M. Kluck (Eds), *Cross-language information retrieval and evaluation*, CLEF 2001. Springer, Berlin, 27-43.

Savoy, J. (2002b). Report on CLEF-2002 experiments: Combining multiple sources of evidence. Proceedings CLEF-2002, 31-46.

Westerveld, T., Kraaij, W. & Hiemstra, D. (2002). Retrieving Web pages using content, links, URLs and anchors. Proceedings TREC-10, NIST Publication #500-250, 663-672.

Appendix 1. Weighting schemes

To assign an indexing weight w_{ij} reflecting the importance of each single-term T_j in a document D_i, we may use the formula shown in Table A.1, where document length (the number of indexing terms) for document D_i is denoted by nt_i, and n indicates the number of documents in the collection. For the Okapi weighting scheme, K represents the ratio between the length of document D_i measured by l_i (sum of tf_{ij}) and the collection's mean is noted by advl or more precisely

$$K = k_1 \left[(1-b) + b \frac{l_i}{avdl} \right]$$

For the Arabic corpus, the constant advl is set at 300, the constant b at 0.55, the constant k_1 at 3. For both Web searching tasks, we set advl at 750, the constant b at 0.9, the constant k_1 at 1.2. For the Lnu scheme, the constant pivot was fixed at 125 and the constant slope at 0.1.

bnn	$w_{ij} = 1$	nnn	$w_{ij} = tf_{ij}$
ltn	$w_{ij} = (\ln(tf_{ij}) + 1) \cdot idf_i$	atn	$w_{ij} = idf_i \cdot [0.5 + 0.5 \cdot tf_{ij} / \max tf_{i.}]$
lnc	$w_{ij} = \frac{\ln(tf_{ij}) + 1}{\sqrt{\sum_{k=1}^{t} \left(\left(\ln(tf_{ik}) + 1 \right) \right)^2}}$	npn	$w_{ij} = tf_{ij} \ln \left[(n - df_j) / df_j \right]$
Okapi	$w_{ij} = \left[(k_1 + 1)\ tf_{ij} \right] / \left(K + tf_{ij} \right)$	dtn	$w_{ij} = \left(1 + \ln \left(1 + \ln(tf_{ij}) \right) \right) idf_j$
ntc	$w_{ij} = \frac{tf_{ij}\ idf_j}{\sqrt{\sum_{k=1}^{t} \left(tf_{ik}\ idf_k \right)^2}}$	dtu	$w_{ij} = \frac{\left(1 + \ln \left(1 + \ln(tf_{ij}) \right) \right) idf_j}{(1 - slope)\ pivot + slope\ nt_i}$
ltc	$w_{ij} = \left[\left(\ln(tf_{ij}) + 1 \right) idf_j \right] / \sqrt{\sum_{k=1}^{t} \left(\left(\ln(tf_{ik}) + 1 \right) idf_k \right)^2}$		
Lnu	$w_{ij} = \frac{\left[\ln(tf_{ij}) + 1 \right] / \left[\ln(nt_i) + 1 \right]}{(1 - slope)\ pivot + slope\ nt_i}$		

Table A.1: Weighting schemes