Report on the TREC-2003 Experiment: Genomic and Web Searches

Jacques Savoy, Yves Rasolofo, Laura Perret

Institut interfacultaire d'informatique, University of Neuchatel (Switzerland)
E-mail: {Jacques.Savoy, Yves.Rasolofo, Laura.Perret}@unine.ch

Summary

This year we took part in the genomic information retrieval and information extraction tasks, as well as the named page and topic distillation searches. In carrying out the last two tasks, we made use of link anchor information and document content in order to construct Web page representatives. This type of document representation uses multi-vectors in order to highlight the importance of both link anchor information and document content.

Introduction

As a part of the TREC-2003 evaluation campaign, the UniNE group is taking part in the genomics and Web tracks. The first section of this paper describes the IR models we used in the genomics information retrieval. Section 2 describes our various approaches to automatically extracting gene descriptions from a given scientific paper, using only its title and its abstract. Section 3 describes our procedures for indexing and retrieving Web pages, based on three document representations, and our distributed indexing framework based on the Okapi probabilistic model. Section 4 explains the IR approach we used to combine both Web page content and anchor information when searching for specific named pages and homepages. Finally, Section 5 describes how our IR scheme can be used within the context of topic distillation tasks.

In order to evaluate our hypothesis, we used the SMART system as a testbed, implementing various vector-space IR schemes and probabilistic models. 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. Genomics Information Retrieval

In carrying out our genomic ad hoc search task, we worked with a Medline collection subset containing 525,938 documents, or more precisely, bibliographic records mainly containing an article title, an abstract and some manually assigned descriptors. The available queries consisted of a gene name (or more precisely, its official name and symbol, together with various alias, associated products and proteins).

This information retrieval task is comparable to the Amaryllis corpus composed of 148,688 scientific bibliographic records written in French. Each of the Amaryllis collection records mainly consisted of an article title, an abstract and some manually assigned descriptors. This collection was included in the CLEF-2002 evaluation campaign, based on 25 requests (Savoy, 2003).

The indexing procedure we used in both genomic tracks is described in Section 1.1, while Section 1.2 provides an overall description of various IR models. Section 1.3 describes the pseudo-relevance feedback (or blind query expansion) used in our experiments. Section 1.4 shows how we combine the various result lists generated, using different indexing and searching schemes in order to process the same document collection (data fusion). The last section evaluates the retrieval effectiveness achieved by diverse IR models and also that of various combined approaches.

1.1. Indexing Procedure

We chose the SMART system as effective means of searching the Medline collection subset. From the original documents and during the indexing process, we retained only the following logical sections: TI (title), TT (transliterated title from non-Roman alphabet language), AB (abstract), MH (MeSH terms based on the NLM's controlled vocabulary), GS (Gene symbol or abbreviated gene names), and RN (EC/RN numbers are assigned by the Enzyme Commission to designate a particular enzyme).

Only the available descriptions were used to formulate the queries. The “official symbol” field was repeated three times however in order to assign more importance to this particular field.

To form an indexing word, our system takes letter and digit sequences into account, or when preceded and followed by a letter, the following character is used: '.@!_ (the default SMART system list). Thus the strings "IBM360", "U.S", or "sym_name" are viewed as single indexing terms. To this set, we might add / and – characters (a set labeled as “Separator+”) in order to view the sequence "DEAD/H" as one indexing term.

Moreover, a stemming procedure was often used in order to reduce to the same root (or stem) inflectional

and derivational variants of words. For example, the words "thinking", "thinkers" or "thinks" would be reduced to the stem "think". To achieve this, we employed the Lovins' stemmer (Lovins, 1968) based on a list of over 260 suffixes (the default stemming approach in SMART). On the other hand, we sometimes preferred using a light stemming approach, wherein only the plural form of English words were removed. To evaluate this stemming approach, we adopted the "S stemmer" (Harman, 1991) based on the following rules:

1. If a word ends in «-ies», but not «-eies» or «-aies» then replace «-ies» by «-y»;
2. If a word ends in «-es», but not «-aes», «-ees» or «-oes» then replace «-es» by «-e»;
3. If a word ends in «-s», but not «-us» or «-ss» then remove the «-s».

1.2. Search Models

In order to define a retrieval model, we must specify how the documents and the requests are to be represented (indexing procedure) and how the similarity between document and query surrogates are to be computed. To achieve this and in order to obtain a broader view of the relative merit of various retrieval models, we evaluated the genomic corpus using 11 search models.

As a first approach, we adopted a binary indexing scheme within which each document (or request) was represented by a set of keywords, without any weight. To measure similarity between documents and requests, we counted the number of common terms, computed according to the inner product (retrieval model denoted "doc=bnn, query=bnn" or "bnn-bnn"). For document and query indexing however binary logical restrictions are often too limiting. In order to weight the presence of each indexing term in a document surrogate (or in a query), we sometimes took account of the term occurrence frequency, thus allowing for better term distinction and increasing indexing flexibility (model denoted "doc=nnn, query=nnn" or "nnn-nnn").

Those terms however that did occur very frequently in the collection were very helpful in distinguishing between relevant and non-relevant items. Thus we could count their frequency in the collection, or more precisely the inverse document frequency (denoted by idf_j), resulting in more weight for sparse words and less weight for more frequent ones. This idf value is usually computed as $\ln(n/df_j)$ where n indicates the number of documents in the collection. Moreover, a cosine normalization could prove beneficial and each indexing weight could vary within the range of 0 to 1 (retrieval model notation: "ntc-ntc"). Table A.1 in the Appendix depicts the exact weighting formulation.

Other variants were also created, especially when considering the occurrence of a given term in a document as a rare event. Thus, it could be good practice to assign more importance to the first occurrence of this word as compared to any successive or repeating occurrences. Therefore, the tf component could be computed as 0.5 + 0.5 [tf / max tf in a document] (retrieval model denoted "doc=atn"). Moreover, we should consider that a term's presence in a shorter document provides stronger evidence than it does in a longer document. To account for this, we integrated document length within the weighting formula, leading to more complex IR models; for example, the IR model denoted by "doc=Lnu" (Buckley *et al.*, 1996), "doc=dtu" (Singhal *et al.*, 1999).

In addition to previous models based on the vector-space approach, we also considered probabilistic models, an example being the Okapi probabilistic model (Robertson *et al.*, 2000). As a second probabilistic approach, we implemented the Prosit (PRObabilistic Sift of Information Terms) approach (Amati & van Rijsbergen, 2002; Amati *et al.*, 2003), based on the following indexing formula:

$$w_{ij} = Inf^1_{ij} \cdot Inf^2_{ij} = (1 - Prob^1_{ij}) \cdot Inf^2_{ij} \quad \text{with}$$
$$Prob^1_{ij} = tfn_{ij} / (tfn_{ij} + 1)$$
$$tfn_{ij} = tf_{ij} \cdot \log_2[1 + ((C \cdot \text{mean dl}) / l_j)]$$
$$Inf^2_{ij} = -\log_2[1 / (1+l_j)] - tfn_{ij} \cdot \log_2[l_j / (1+l_j)]$$
$$\text{with } l_j = tc_j / n$$

in which w_{ij} reflects the importance of each single-term t_j in a document D_i, tf_j the frequency of occurrence of term t_j in a document D_i, tc_j indicates the number of occurrences of term t_j in the collection, n the number of documents, and C and mean dl are constants. In this model, the query terms are weighted according to a term occurrence frequency (denoted "nnn").

1.3. Pseudo-Relevance Feedback

It was observed that pseudo-relevance feedback (or blind-query expansion) seemed to be a useful technique for enhancing retrieval effectiveness. In our evaluations, we adopted Rocchio's approach (Rocchio, 1971), (Buckley *et al.*, 1996) in which the newly expanded query Q' was composed as follows:

$$Q' = \alpha \cdot Q + \beta \cdot \frac{1}{r} \cdot \sum_{j=1}^{r} w_{ij}$$

within which Q denoted the previous request, $\alpha = 0.75$, $\beta = 0.75$ and the system was allowed to add s terms extracted from the original query's r best ranked documents.

	Mean average precision			
	Lovins' stemmer		S stemmer	
IR model	Default list	Separator+	Default list	Separator+
Okapi-npn	15.10	15.39	15.53	15.41
OkaR-npn	14.74	15.13	15.39	15.45
Prosit	15.07	14.60	15.31	14.91
dtu-dtn	**16.80**	**17.44**	**18.25**	**18.67**
atn-ntc	16.47	16.30	16.71	16.86
Lnu-ltc	16.19	16.28	16.62	16.41
ltn-ntc	16.32	16.39	16.24	16.00
ltc-ltc	15.64	15.95	16.62	16.43
lnc-ltc	14.86	14.00	16.00	14.64
ntc-ntc	14.48	12.78	15.04	13.30
bnn-bnn	7.02	7.48	7.31	8.54
nnn-nnn	3.90	3.54	3.48	3.84

Table 3. Mean average precision for various IR models using the Genomics corpus (50 queries)

1.4. Data Fusion

Until now, we used a single search model (or engine) when searching document collections. We might however suggest sending the request to various search engines that would handle the same document collection but that use different indexing or search schemes. Once we have obtained result lists from these various search engines, we would need to merge them in an effective manner (data fusion). Thus, even though certain degrees of retrieval effectiveness may be attributed to each search approach, combining the result lists might provides 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) suggested using various operators (see Table 1) and showed that the best performance could 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 1. Data fusion strategies

Of course we might also employ the round-robin merging strategy, taking 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 result list, thus providing the next item to be put in the final list. Duplicates encountered in this process are simply ignored.

1.4. Evaluation

To evaluate various IR models using the genomic collection, we used 50 queries and various statistics on relevance assessments. As Table 2 illustrates, the number of pertinent items per request is relatively small (mean = 11.32). The mean average precision for nine vector-space schemes together with the Okapi and Prosit probabilistic models are depicted in Table 3. This table shows the results of evaluating two stemming approaches and two word delimiting strategies.

Number of queries	50
Number of relevant doc.	566
Mean rel. doc. / request	11.32
Standard deviation	13.15
Median	7
Maximum	66 (Query #32)
Minimum	2 (e.g., Query #4)

Table 2. Relevance judgment statistics (Genomics)

An examination of Table 3 shows that the best retrieval effectiveness was obtained when using the vector-space model "dtu-dtn," while second best results were usually obtained using the "atn-ntc" scheme. Ranking third was the "Lnu-ltc" model (using the S stemmer), or the "ltn-ntc" model (using Lovins' stemmer). In these experiments, the simple "tf-idf" approach (denoted "ntc-ntc") did not appear to perform very well. To our surprise, we noted that the Okapi or Prosit probabilistic model did not perform very well in this task, contrarily to our previous experiments (for example those based on the Amaryllis corpus, also composed of bibliographic records (Savoy, 2003)).

	Mean average precision							
	Lovins' stemmer				S stemmer			
	Default list		Separator+		Default list		Separator+	
IR model	Prosit	dtu-dtn	Prosit	dtu-dtn	Prosit	dtu-dtn	Prosit	dtu-dtn
#doc/#term	15.07	16.80	14.60	17.44	15.31	18.25	14.91	**18.67**
3 / 10	15.60	16.77	15.55	17.53	16.25	18.57	16.43	16.27
3 / 15	15.62	16.45	15.63	17.50	16.26	18.40	16.44	15.88
3 / 20	15.58	16.59	15.80	17.42	15.93	18.47	16.55	16.03
3 / 30	16.10	16.51	15.41	17.46	16.41	18.36	16.92	15.97
5 / 10	16.14	16.75	16.42	17.55	16.79	18.57	16.78	16.27
5 / 15	16.43	16.78	15.65	17.60	16.56	18.61	16.46	16.47
5 / 20	16.30	16.74	15.69	17.62	16.03	18.57	16.87	16.14
5 / 30	15.50	16.76	15.31	17.61	15.87	18.58	16.29	16.05
10 / 10	**17.17**	16.74	17.20	17.67	**17.90**	18.50	17.37	15.93
10 / 15	16.53	16.75	16.93	17.67	16.97	18.57	**17.53**	16.05
10 / 20	15.96	16.76	**17.38**	17.67	16.79	**18.62**	17.29	16.20
10 / 30	15.97	**16.79**	16.37	**17.68**	16.10	18.42	16.81	16.35

Table 4. Mean average precision for various relevance feedback parameter settings (Genomics corpus, 50 queries)

	Mean average precision							
	Lovins' stemmer				S stemmer			
	Default list		Separator+		Default list		Separator+	
IR model		+ Q expand		+ Q expand		+ Q expand		+ Q expand
Prosit	15.07	17.17	14.60	17.38	15.31	17.90	14.91	17.53
dtu-dtn	16.80	16.79	**17.44**	17.68	18.25	18.62	**18.67**	16.35
combMAX	15.07	17.17	14.60	17.38	15.31	17.90	14.91	17.53
combMIN	2.48	1.46	5.02	1.26	3.06	1.34	5.16	2.05
combSUM	15.51	17.08	15.47	17.43	15.72	16.89	16.11	17.52
combANZ	12.16	14.00	12.33	13.39	13.79	15.76	14.15	15.27
combNBZ	15.51	17.07	15.48	17.42	15.72	17.90	16.11	17.39
combRSV%	17.43	17.52	16.52	17.63	18.38	18.72	17.82	16.97
combRSVn	**17.54**	17.51	16.78	17.72	**18.48**	**18.74**	17.83	17.07
round-robin	15.66	**17.53**	16.05	**18.21**	16.67	18.59	16.97	**17.59**

Table 5. Mean average precision of various data fusion approaches (Genomics corpus, 50 queries)

From Table 3 we might also infer that the extended word delimiter (labeled "Separator+") usually enhanced performance only slightly. Moreover, the light S stemmer resulted in better retrieval effectiveness than did Lovins' algorithm.

From the data depicted in Table 4, we can conclude that pseudo-relevance feedback usually increases mean average precision. When taking the r=10 best ranked documents into account, performance is usually enhanced compared to r=3 or r=5. This improvement is however rather small, particularly for the "dtu-dtn" vector-space model. On the other hand from previous experiments with the Prosit model, there is evidence that blind query expansion usually improves mean average precision significantly (Savoy, 2003). Our current test-collection seems to confirm this. Finally, we evaluated various data fusion strategies that might be employed to improve retrieval effectiveness. In our case we submitted the same request to two search engines (Prosit and "dtu-dtn") with and without blind query expansion (using the best parameter setting). Based on the data shown in Table 5, it appears that data fusion based on combRSVn or a simple round-robin scheme performs better. Moreover, various data fusion strategies (combMIN, combMAX, combANZ, and combNBZ) degraded the system's overall performance.

Table 6 shows the results of combining the Prosit and "dtu-dtn" search models, using both stemmers (denoted "Lov" or "S") and separator characters lists (our separator list "Separator+" is denoted "+"). From this data, we can conclude that it seemed better to combine retrieval schemes based on a variety of

indexing strategies (e.g., using the different separator lists shown in the second column, or different stemming algorithms as depicted in the third and forth columns). Finally, Table 7 lists the specifications for our official runs.

Prosit dtu-dtn	17.90 (S) 18.67 (S+)	17.38 (Lov+) 18.62 (S)	17.38 (Lov+) 18.67 (S+)
combMAX	18.49	17.38	16.98
combMIN	15.97	1.21	15.00
combSUM	19.17	17.51	18.69
combANZ	18.73	13.01	18.38
combNBZ	19.09	17.49	18.63
combRSV%	**19.45**	18.81	**19.18**
combRSVn	19.44	**18.91**	19.15
round-robin	19.09	18.81	17.16

Table 6. Evaluation of various data fusion strategies

Run name	MAP	Description
UniNEg1	18.52	Okapi+dtu-dtn, def., combRSVn
UniNEg2	18.02	Okapi+dtu-dtn, def., combRSV%
UniNEg4	16.23	dtu-dtn, Lovins, default list
UniNEg5	16.35	Lnu-ltc, S-stem, separator+

Table 7. Description of our official runs
(all with blind query expansion)

2. Genomic Information Extraction

The main purpose of the genomic secondary task was to address the bioinformatic community's information extraction needs. More precisely, the goal was to reproduce the GeneRIF (Gene Reference into Function used in the LocusLink[1] database), either from a Medline record or from the entire article. GeneRIF snippets sometimes contain direct quotations from article abstracts but they might also include or paraphrase certain texts extracted from article titles or abstracts.

The data used for this task consisted of 139 GeneRIFs, representing all articles appearing in five journals (*Journal of Biological Chemistry*, *Journal of Cell Biology*, *Science*, *Nucleic Acids Research* and *Proceedings of the National Academy of Sciences*), during the latter half of 2002.

2.1. Models

From the beginning, we decided to use only the article titles and abstracts for this task. As the title was supposed to be a good candidate for the GeneRIF annotation, we tried selecting it systematically and using it as a baseline performance measure for our task.

Then for each GeneRIF we tried selecting each GeneRIF term also contained in the corresponding abstract. This method provided us with a theoretical maximum that could be reached, using only articles titles and abstracts.

2.1.1. Dummy (UniNEie5)

First of all, we established the term frequencies for the words contained in the GeneRIFs. Then, we ranked them and selected in descending order, those terms having frequencies greater or equal to 9. The words selected by using this simple strategy were:

cell role protein expression gene receptor activation regulate human apoptosis alpha sp1 signaling domain regulation kinase suggest pathway

We then supplied this fixed sequence of words as the GeneRIF for each query.

2.1.2. Random (UniNEie4)

For each query, we segmented the corresponding abstract into sentences. Then we considered all sentences, including the title. Each sentence having 10 to 14 words was repeated once into our set of candidates. Each sentence within this set thus had an equal probability of being selected. Finally we randomly chose a sentence that was returned as the GeneRIF.

2.1.3. Logarithm of Term Frequency (UniNEie3)

As the GeneRIFs were provided, we computed the term frequencies for all words contained therein. Then, for each query, we segmented the corresponding abstract into sentences. For each sentence, including the title, we removed the stopwords and then stemmed the remaining words, using the SMART stopword list (571 entries) and the S stemmer (see Section 1.1). We then computed a sentence score as follows:

$$\text{score} = \sum_{j=1}^{len} \ln(tf_j) \Big/ len$$

where j is the term index, tf_j the term frequency in GeneRIFs and len the length of the sentence without stopwords. Finally, we returned the sentence having the highest score as the GeneRIF.

2.1.4. Term Frequency and Logistic Regression

We again used the above process except that the score was computed as follows:

[1] Available at www.ncbi.nlm.nih.gov/LocusLink/

$$score = \sum_{j=1}^{len} w(tf_j) / len$$

where j is the term index, tf_j the term frequency in GeneRIFs, and len is the length of the sentence without stopwords. Table 8 lists the resultant $w(tf_j)$ values.

We then selected the sentence having the highest score as a GeneRIF candidate and applied a logistic regression model (Hosmer & Lemeshow, 2000), using the statistical Fisher method to predict when the system should return the chosen sentence or the title. We tried two variants, corresponding to different sets of variables.

tf_j	$w(tf_j)$
$9 < tf_j$	4
$4 < tf_j \leq 9$	3
$2 < tf_{ij} \leq 4$	2
$1 < tf_{ij} \leq 2$	1
$tf_{ij} \leq 1$	0

Table 8. Terms weights

Model A (UniNEie2)

The following example provides an explanation of how our model works. Looking at Query #30, we must chose between the title and the candidate shown in Table 9. Table 10 lists the sentence results after removing stopwords and applying the stemming procedure.

Title	Comparative surface accessibility of a pore-lining threonine residue (T6') in the glycine and GABA(A) receptors.
Candidate	This action was not induced by oxidizing agents in either receptor.

Table 9. Competing sentences for Query #30

For each candidate, we could compute certain statistics, such as its length ("Len"), the number of acronyms ("Abrv"), the number of indexing terms ("Terms"), etc. as shown in Table 11. Since however we knew the title can usually be viewed as a suitable GeneRIF, we also computed certain statistics concerning the difference between a given candidate and the article title, as shown in Table 12. These values were then used as predictors in our logistic regression model to compute the probability that the corresponding candidate would be a suitable GeneRIF. The last column in Table 12 lists the estimated value of these corresponding statistics. For example, the estimate for the variable "d.Len" is negative, indicating that when the candidate length is greater than the title length, this fact decreases the probability that this candidate would be a suitable GeneRIF.

Title	Comparative surface accessibility pore-lining threonine residue (T6') glycine GABA(A) receptor
Candidate	action induced oxidizing agent either receptor

Table 10. Competing sentences (stemmed, without stopwords)

Variable	Meaning	Candidate	Title	Diff
Len	length	6	10	-4
Abrv	#acronyms	0	1	-1
Terms	#terms	5	10	-5
Max2Idf	2nd max idf	3.44	9.01	-5.58
MinIdf	min idf	2.25	2.35	-0.11
Min2Idf	2nd min idf	2.65	2.65	0.0

Table 11. Variables used for the regression

Variable	Meaning	Estimate
Intercept		-3.9502
d.Len	length candidate – title	-3.3939
d.Abrv	# acronyms candidate - title	2.5182
d.Terms	# terms candidate - title	2.5645
d.Max2Idf	2nd max idf candidate - title	1.1829
d.MinIdf	min idf candidate - title	6.1737
d.Min2Idf	2nd min idf candidate - title	-5.1732

Table 12. First set of variables and estimates

Using the result of the logistic regression, we returned the complete title 126 times and the candidate 13 times, 7 of them forming a part of the title.

Model B (UniNEie1)

As a variant of the previous model, we changed the set of explanatory variables, as depicted in Table 13.

Using the logistic regression results, we returned the complete title 129 times and our candidate 10 times, 6 of them forming a part of the title.

2.2. Evaluation

The Dice coefficient, measuring the degree of overlap of two sentences was used for evaluation purposes. Given two sentences A and B, we defined |A| as the number of words in A, |B| as the number of words in B, and $|A \cap B|$ as the number of words

occurring in both A and B. The Dice coefficient was measured by:

$$\text{Dice}(A, B) = \frac{\left(2 * |A \cap B|\right)}{\left(|A| + |B|\right)}$$

Four variants of this measure were used for evaluation (more details can be found at the Web site[1])

- Dice 1 is the classical Dice
- Dice 2 is the modified unigram Dice
- Dice 3 is the bigram Dice
- Dice 4 is the bigram Phrases

Variable	Meaning	Estimate
Intercept		68.199
Terms	# indexing terms in the candidate	-19.867
Min2Idf	2nd max idf candidate	-36.733
nb.Art	# common terms in candidate and abstract	18.999
d.Len	length candidate –title	-57.029
d.Abrv	# acronyms candidate - title	17.141
d.Terms	# indexing terms candidate - title	46.910
d.Max2Idf	2nd max idf candidate - title	30.926
d.MinIdf	min idf candidate - title	22.121

Table 13. Second set of variables and estimates

Table 14 shows an evaluation of our runs. The second row forms our baseline, representing the article title, a scheme within which the title is always returned. On the other hand, the third row ("Generifs ∩ abst.") represents the maximum value that could be achieved when selecting the most appropriate sentence, using only the article title and abstract.

	Dice 1	**Dice 2**	**Dice 3**	**Dice 4**
Title (min)	50.47%	52.60%	34.82%	37.91%
Generifs ∩ abst. (max)	59.53%	83.26%	61.66%	52.76%
UniNEie5	9.42%	14.20%	0.15%	0.17%
UniNEie4	25.88%	25.29%	12.03%	13.61%
UniNEie3	49.46%	51.42%	33.62%	36.99%
UniNEie2	51.72%	54.27%	36.62%	39.71%
UniNEie1	**52.28%**	**54.78%**	**37.43%**	**40.35%**

Table 14. Evaluation of our official runs

[1] See medir.ohsu.edu/~genomics/protocol.html

Using this data, we hoped to improve our extraction of the suitable GeneRIFs from the title scheme, through using one of our logistic regression models. In this case, it seemed that Model B (or UniNEie1) performed slightly better, even though both models returned the title many times. We attempted to improve our system's performance through incorporating additional data, such as full text articles or gene names, together with the selection of the explanatory variable set for the logistic regression.

3. Our Okapi Search Model

Based on our previous work (Savoy & Picard, 2001; (Savoy & Rasolofo, 2003), the Okapi search model provided significantly greater retrieval effectiveness. However, in order to manage the Web collection (1,247,753 documents that were extracted from the .GOV domain, or about 18.1 GB of data), we needed to modify this search model for two reasons. Firstly, we wanted to incorporate three document representatives for each Web page, and secondly we needed to distribute the inverted file in order to respect the 2 GB limit.

When processing three document representations, we estimated the degree of similarity between document D_i and the current query would be a linear combination of the inner product of the three document representations, to be given as:

$$RSV(D_i) = \alpha \cdot \sum_{j=1}^{m} w_{ij}^{(1)} \cdot qw_j + \beta \cdot \sum_{j=1}^{m} w_{ij}^{(2)} \cdot qw_j + \gamma \cdot \sum_{j=1}^{m} w_{ij}^{(3)} \cdot qw_j \quad (1)$$

where $w^{(1)}_{ij}$ indicates the weight attached to the term t_j in the document D_i in the first document representation ($w^{(2)}_{ij}$ and $w^{(3)}_{ij}$ for the second, respectively, the third document surrogate), and the parameters α, β, γ are used to assign a comparative importance to each document representative.

Creating a single inverted file from a collection of approximately 18 GB might be impossible using a 32-bit system (e.g., Linux). To overcome this limit, we will follow the approach described in Rasolofo & Savoy (2003), whereby each sub-collection would be indexed using the tf component. When merging the result lists obtained from searching into these different sub-collections, we computed the idf component and applied the normalization. Following this step, we could then merge the result lists according to the new document scores.

Knowing that both Web tasks required high precision searches, we decided to enhance our Okapi

model by implementing the term proximity scoring function (see Rasolofo & Savoy (2003) for more details). The main premise was that if all search keywords appear in a document representative, our search model would increase the corresponding document score. On the other hand, if only a part of these search terms appeared in a given Web page, the final retrieval status value would remain unchanged (see Eq. 1). While the term proximity function would have a greater value if the search keywords appear close to each other, they may occur more than once within a given sentence or tag. In our system the constant δ denoted the impact of these proximity scores. Of course, setting $\delta = 0$ means that the proximity score is not computed.

4. Named and Home Pages Finding

The following considerations formed the basis of our first Web task. When submitting a request to a search engine, users will sometimes not want a ranked list of Web pages concerning a particular topic, but rather they would prefer the location of an underlying service or known-item (usually presented within a short list of the most probable locations). For example, the appropriate response to a query on "state department", "Secret Service jobs", "Navajo Nation", or "barbara mikulski bio" (and even with a spelling error such as "US Volcano Oservatories") would not be a ranked list of documents covering these subjects but rather those site(s) that contain the required form/information/list. To accomplish this we needed to implement an IR system that could retrieve a limited number of pages (one at the very least) in response to the user's request.

4.1. Search Models

As a basis for our search model we used the Okapi model as described in Section 3. Our first document representative was based on information found in the Web page, including the corresponding <TITLE> and <META> tags ("keywords" and "description"). Of course Web pages might also contain links and their associated anchor texts (or anchor texts for outgoing links). Our second document surrogate was based on the <TITLE> tag and the anchor texts for those Web pages pointing to the current document. The third document representative was built by concatenating the <TITLE>, <H1>, and <BIG> tags from pages pointing to the current Web page. This third aspect was used to reinforce the importance of those Web pages pointing to the current page. Since we know that end tags (e.g., </H1> or </BIG>) are sometimes missing, we only considered the first 64 words following any given tag.

This indexing strategy was based on previous studies (Craswell *et al.*, 2001), (Westerveld *et al.*, 2002), (Kraaij *et al.*, 2002), showing that anchor texts from other Web pages pointing to the current page may provide compact and often accurate descriptions of the current page's content. For this reason, we extracted link anchor texts from all Web pages pointing to the current page and concatenated them to form our second document representative. Finally, we also considered URL content (or more precisely, the similarity between the URL text and the current request, or URL lengths). In our current search models, these additional sources of information had not taken into account.

When high precision results are required for indexing documents or requests, it is usually not a good idea to include a stemming procedure. We could however adopt a light stemming such as the "S-stemmer" (see Section 1.1 (Harman, 1991)). In this case, the words "house" and "houses" would be reduced to the same root while the term "housing" would be treated as a different indexing unit. Based on our experiments from last year (Savoy & Rasolofo, 2003), we decided to ignore the stemming approach for this task due to the fact that even light stemming was usually found to diminish the system's overall performance (Savoy & Rasolofo, 2003).

4.2. Evaluation

In this IR search model, based on three document representatives and a proximity scoring function, we first needed to determine the relative importance assigned to each document representative (based on internal Web page content for the first surrogate), as compared to the weight attached to the second and third document representatives (based mainly on link anchor texts from those Web pages pointing to the current one). The relative importance for each surrogate was controlled through using the parameters α, β, γ (see Section 3) while the proximity score was weighted using the constant δ.

Number of queries	300
Number of relevant doc.	352
Mean rel. doc. / request	1.173
Standard deviation	0.609
Median	1
Maximum	6 (Query #244)
Minimum	1

Table 15. Relevance judgment statistics (named and home page searches, TREC-2003)

Our evaluation was based mainly on the mean reciprocal rank (MRR) of the first correct answer found by the system. Table 15 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 300 queries, we considered only the first 100 retrieved items. As seen in Table 16, the best value for our parameters seems to be around α=0.6, β=0.4, γ=0.05, and δ=0.1, thus assigning a little more weight to internal representation (parameter α) than to the anchor texts of all Web pages pointing to the current document (parameter β). The third representation does not seem to have a great impact on system's performance. The underlined parameters in this table represent the settings used for our official runs.

Parameters	MRR	# in top 10
Okapi b=0.5		
α=0.6, β=0.4, γ=0, δ=0	0.666	252 (84.0%)
α=0.6, β=0.4, γ=0, δ=0.1	0.691	251 (83.7%)
α=0.6, β=0.4, γ=0, δ=0.2	0.692	251 (83.7%)
α=0.6, β=0.4, γ=0.05, δ=0	0.707	258 (86.0%)
α=0.6, β=0.4, γ=0.05, δ=0.1	**0.720**	259 (86.3%)
α=0.7, β=0.3, γ=0.05, δ=0.1	0.700	258 (86.0%)
α=0.8, β=0.2, γ=0.05, δ=0.1	0.676	252 (84.0%)
α=0.8, β=0.2, γ=0.05, δ=0.2	0.682	254 (84.7%)
Okapi b=0.6		
α=0.6, β=0.4, γ=0, δ=0	0.667	250 (83.3%)
α=0.6, β=0.4, γ=0, δ=0.1	0.690	250 (83.3%)
α=0.6, β=0.4, γ=0, δ=0.2	0.689	250 (83.3%)
α=0.6, β=0.4, γ=0.05, δ=0	0.700	258 (86.0%)
α=0.6, β=0.4, γ=0.05, δ=0.1	**0.713**	259 (86.3%)
α=0.7, β=0.3, γ=0, δ=0	0.626	247 (82.3%)
α=0.7, β=0.3, γ=0, δ=0.1	0.658	251 (83.7%)
α=0.7, β=0.3, γ=0.05, δ=0.1	0.699	257 (85.7%)
α=0.8, β=0.2, γ=0.05, δ=0.1	0.686	254 (84.7%)
Okapi b=0.7		
α=0.6, β=0.4, γ=0, δ=0	0.654	246 (82.0%)
α=0.6, β=0.4, γ=0, δ=0.1	0.677	247 (82.3%)
α=0.6, β=0.4, γ=0, δ=0.2	0.676	248 (82.7%)
α=0.6, β=0.4, γ=0.05, δ=0	0.691	256 (85.3%)
α=0.6, β=0.4, γ=0.05, δ=0.1	0.701	256 (85.3%)
α=0.6, β=0.4, γ=0.05, δ=0.2	**0.706**	257 (85.7%)
α=0.7, β=0.3, γ=0.05, δ=0.1	0.688	254 (84.7%)
α=0.8, β=0.2, γ=0.05, δ=0.1	0.683	253 (84.3%)

Table 16. IR model evaluation for various combinations of our three document representatives

Finally, Table 17 provides a summary description of our four official runs. Usually, we did not attach much importance (δ = 0 or very small) to the third document representative (<TITLE>, <H1>, and <BIG> texts from pages pointing to the current Web page). The difference between UniNEnp1 and UniNEnp3 represented the inclusion of the term proximity scoring function within UniNEnp3, seemingly a useful technique for improving retrieval effectiveness. Taking account for this third surrogate enhanced the system's performance (see Table 16). The performance differences between UniNEnp2, UniNEnp5 were due to the various parameter settings used for the Okapi model.

Run name	MRR	Parameter settings
UniNEnp1	0.626	Okapi b=0.6 α=0.7, β=0.3, γ=0.0, δ=0.0
UniNEnp2	0.676	Okapi b=0.5 α=0.8, β=0.2, γ=0.05, δ=0.1
UniNEnp3	0.658	Okapi b=0.6 α=0.7, β=0.3, γ=0.0, δ=0.1
UniNEnp4	**0.688**	Okapi b=0.7 α=0.7, β=0.3, γ=0.05, δ=0.1
UniNEnp5	0.686	Okapi b=0.6 α=0.8, β=0.2, γ=0.05, δ=0.1

Table 17. Description of official named-page & homepage runs

5. Topic Distillation

The basic purpose of the topic distillation task was to return a list of key resources on a given topic (e.g., "pest control safety", "computer viruses" or "children's literature"). Explicitly defining what does or does not constitute a suitable resource was however difficult, and each definition seemed to become more and more ambiguous. While Web pages with appropriate content might be considered as useful key resources and we could have retrieved them using a classic IR model, key resources may also be good hubs (or Web pages pointing to various pages containing pertinent content with respect to the submitted request). Moreover, if a Web page is linked to two, three or more children having a high degree of similarity with the request, it seems more appropriate to return this parent page rather than the two, three of more children. 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 various strategies that would point to reliable browsing starting points rather than simply retrieving Web pages with suitable content.

5.1. Search Models

As for the named page and homepage search task, we built three 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") and the anchor texts contained in the page. The second document surrogate was built from the text

delimited by the <TITLE> tag together with link anchor texts from all outgoing and all incoming links. The third document representative was composed of all <TITLE> and <H1> tags provided by all pointed pages (or pages accessible within a one-click distance from the current page).

Once the pages were 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 direction). To retrieve these starting points we used our spreading activation (SA) search scheme (Savoy, 1996), (Crestani & Lee, 2000), (Savoy & Picard, 2001). Using this method, document scores initially computed by the IR system (denoted $RSV(D_i)$) were propagated to the linked documents through a certain number of cycles, based on a propagation factor. We used a simplified version with only one cycle and a fixed propagation factor λ for all links. As a result, the final retrieval status value for a document D_i linked to k documents was computed using the following equation:

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

When in our experiments we tried to extract the proper starting sites for browsing, we only considered all incoming links for each of the k best-ranked documents.

As other possibilities, we might consider the PageRank algorithm (Brin & Page, 1998), the HITS algorithm (Kleinberg, 1998) or probabilistic argumentation systems (Picard, 1998). During the evaluation campaign of last year however, we did obtain poor performance when employing the HITS algorithm (Savoy & Rasolofo, 2003).

5.2. 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 516 for the 50 queries included in the .GOV collection). Each request would be composed of a short title and a descriptive part.

Table 18 shows various statistics based on relevance assessments. The mean number of relevant items (or key resources) per request is 10.32. From considering the number of distinct roots (e.g., the first part of an URL, e.g., "trec.nist.gov"), we found that in mean, there were 8.38 different roots per query (for Query# 13, the unique relevant item is coming from the Web site "nimh.nih.gov"). On the other hand, for Query# 48, we found 9 relevant pages (over a total of 10) extracted from the root page "prime.jsc.nasa.gov".

Number of queries	50
Number of relevant doc.	516
Mean rel. doc. / request	10.32
Standard deviation	13.38
Median	8
Maximum	86 (Query: #32)
Minimum	1 (Query: #13)
Number of distinct roots / query	
Mean	8.38
Standard deviation	11.641
Median	6
Maximum	77 (Query: #32)
Minimum	1 (Query: #13)
URL length 1	79
length 2	93
length 3	171
length 4	108
length 5	44
length 6	13
length 7 and more	8

Table 18. Relevance judgment statistics (topic distillation searching task, TREC-2003)

In our first set of experiments, we evaluated our extended Okapi IR model (see Section 3). By varying the value attached to the parameters α, β, γ, we assigned more or less weight to each document representation. For example, when we set α to 0, and β to 0, we accounted for text delimited by the <TITLE> and <H1> tags provided by all pointed pages. In other words, we viewed the page as a good starting point for browsing (limited however to a one-click distance). On the other hand, with $\alpha = 1$, $\beta = 0$, and $\gamma = 0$, our search model was based only on Web page content.

Table 19 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 parameter values seemed to be: $\alpha = 0.5$, $\beta = 0.5$, and $\gamma = 0$. The third document representative does not seem to improve retrieval effectiveness. As such, our second representation (<TITLE> tags & anchor link texts from all pointed and pointing pages) seemed to be more valuable for this specific IR task. Usually, the term proximity scoring function seems to improve the ranking of pertinent items (e.g., precision after 5). The underlined parameters in this table represent the settings used for our official runs.

Table 20 provides a summary description of our five official runs, all of which were created without a stemming procedure. For both UniNEdi2 and UniNEdi5, we only accounted for a single document representative

(content-oriented only, based on the good performance of such indexing schemes last year). For UniNEdi3 and UniNEdi4, we accounted for two document surrogates. Our best run was UniNEdi1, which accounted for three document representatives. For UniNEdi4, after retrieving content-based Web pages using our extended Okapi model, we applied a spreading activation with α = 0.02, for the first k = 50 top-ranked items.

Run name	Prec@5	Prec@10
β=1, γ=0, δ=0, ε=0	8.00	6.20
β=1, γ=0, δ=0.03, ε=0.1	11.60	7.60
β=1, γ=0, δ=0.03, ε=0.3	12.40	8.00
β=0, γ=0, δ=1, ε=0	7.20	4.60
β=0, γ=0, δ=1, ε=0.1	8.00	4.60
β=0.5, γ=0.5, δ=0, ε=0	16.40	10.80
β=0.5, γ=0.5, δ=0, ε=0.1	16.00	11.00
β=0.5, γ=0.5, δ=0.03, ε=0	16.40	10.80
β=0.5, γ=0.5, δ=0.03,ε=0.1	16.00	11.00
β=0.5, γ=0.5, δ=0.1, ε=0	15.60	11.40
β=0.5, γ=0.5, δ=0.1, ε=0.1	16.00	**11.60**
β=0.7, γ=0.3, δ=0, ε=0	15.20	10.20
β=0.7, γ=0.3, δ=0, ε=0.1	16.00	10.20
β=0.7, γ=0.3, δ=0.1, ε=0	14.00	11.40
β=0.7, γ=0.3, δ=0.1, ε=0.1	14.00	11.40
β=0.8, γ=0.2, δ=0, ε=0	14.40	9.80
β=0.8, γ=0.2, δ=0, ε=0.1	14.80	9.40
β=0.8, γ=0.2, δ=0, ε=0.3	15.20	9.00
β=0.8, γ=0.2, δ=0.03,ε=0.3	15.20	9.00
β=0.8, γ=0.2, δ=0.1, ε=0.3	14.00	10.60

Table 19. Evaluation of various document surrogates combinations

Run name	Prec@10	description
UniNEtd1	**9.80**	β=0.8, γ=0.2, δ=0.03, ε=0
UniNEtd2	7.60	β=1, γ=0, δ=0, ε=0
UniNEtd3	7.60	β=1, γ=0, δ=0.03, ε=0.1
UniNEtd4	8.80	β=1, γ=0, δ=0.03, ε=0.3 & SA, k=50, α=0.02
UniNEtd5	9.60	β=1, γ=0, δ=0, ε=0 & data fusion

Table 20. Description of our official topic distillation runs

When evaluating our spreading activation (SA) method, we only take account for hyperlinks in reverse orientation. In this case, a αfraction of the score attached to the children is propagated to the parent page (see Eq. 3). From data depicted in Table 21, it seems that the propagation factor α must be around 0.02, and the SA must be limited to the first k = 300 or first k = 400 best-ranked items.

When using the best-run shown in Table 19, we tried various parameter settings as depicted in top part of Table 21. In this case, we may enhance the precision after 10 documents from 11.8% to 14.0% (leading to +20% improvement). On the other hand, when the starting point is based only on the Web page content (as depicted in the bottom part of Table 21, with β= 1, γ = 0, δ = 0, ε = 0), our SA may also improve the precision at 10 retrieved item from 6.2% to 10.2% (+64% improvement).

Parameters	Prec@5	Prec@10
β=0.5, γ=0.5, δ=0.1, ε=0.1	16.00	11.60
α = 0.02, k = 50	17.60	12.00
α = 0.05, k = 50	**17.60**	12.20
α = 0.1, k = 50	13.60	12.00
α = 0.05, k = 100	18.80	12.80
α = 0.05, k = 200	19.20	12.40
α = 0.05, k = 300	16.40	**14.00**
α = 0.05, k = 400	16.00	13.80
β=1, γ=0, δ=0, ε=0	8.00	6.20
α = 0.02, k = 50	10.00	7.20
α = 0.05, k = 50	10.40	7.80
α = 0.1, k = 50	10.00	7.60
α = 0.05, k = 100	12.80	8.60
α = 0.05, k = 200	12.80	9.20
α = 0.05, k = 300	12.80	9.80
α = 0.05, k = 400	13.20	10.20

Table 21. Evaluation of various parameter settings for the spreading activation approach

Acknowledgments

The authors would like to thank C. Buckley from SabIR for allowing us the opportunity to use the SMART system. This research was supported by the SNSF (grant 21-66'742.01).

References

Amati, G. & van Rijsbergen, C.J. (2002). Probabilistic models of information retrieval based on measuring the divergence from randomness. *ACM TOIS*, 20(4), 357-389.

Amati, G., Carpineto, C. & Romano, G. (2003). Italian monolingual information retrieval with PROSIT. In C. Peters, M. Brachler, J. Gonzalo, M. Kluck (Ed.), Cross-Language Information Retrieval and Evaluation. Springer, Berlin.

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.

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.

Hosmer, D.W. & Lemeshow, S. (2000). Applied Logistic Regression. 2nd edn., John Wiley.

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.

Lovins, J.B. (1968). Development of a stemming algorithm. *Mechanical Translation and Computational Linguistics*, 11(1), 22-31.

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, Springer, Berlin, 207-218.

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

Rocchio, J.J. Jr. (1971). Relevance Feedback in Information Retrieval. In *The SMART Retrieval System - Experiments in Automatic Document Processing*, G. Salton (Ed.), Prentice-Hall Inc., 313-323.

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. (2003). Report on the TREC-11 Experiment: Arabic, Named Page and Topic Distillation Searches. Proceedings TREC-11, NIST publication #500-251, 765-774.

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. (2003). Report on CLEF-2002 Experiments: Combining Multiple Sources of Evidence. In C. Peters, M. Brachler, J. Gonzalo, M. Kluck (Ed.), Cross-Language Information Retrieval and Evaluation. Springer, Berlin, to appear.

Singhal, A., Choi, J., Hindle, D., Lewis, D.D., & Pereira, F. (1999). AT&T at TREC-7. Proceedings TREC-7, Gaithersburg: NIST Publication #500-242, 239-251.

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, the formula shown in Table A.1 may be used, 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 avdl or more precisely

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

For the Genomic corpus, the constant avdl was fixed at 300, b at 0.55, k_1 at 1.2, C at 3, mean dl at 73, pivot at 50 and slope at 0.05. For both Web searching tasks, we set avdl at 900, b at 0.75, k_1 at 1.2, pivot 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	$w_{ij} = tf_{ij} \cdot \ln\left(\left(n - df_j\right) / df_j\right)$
Okapi	$w_{ij} = \left((k_1 + 1) \cdot tf_{ij}\right) / \left(K + tf_{ij}\right)$	dtn	$w_{ij} = \left(1 + \ln\left(1 + \ln(tf_{ij})\right)\right) \cdot idf_j$
ntc	$w_{ij} = \frac{tf_{ij} \cdot idf_j}{\sqrt{\sum_{k=1}^{t} \left(tf_{ik} \cdot idf_k\right)^2}}$	dtu	$w_{ij} = \frac{\left(1 + \ln\left(1 + \ln(tf_{ij})\right)\right) \cdot idf_j}{(1 - slope) \cdot pivot + slope \cdot nt_i}$
ltc	$w_{ij} = \left(\ln(tf_{ij}) + 1\right) \cdot idf_j / \sqrt{\sum_{k=1}^{t} \left(\left(\ln(tf_{ik}) + 1\right) \cdot idf_k\right)^2}$		
Lnu	$w_{ij} = \frac{\left(\ln(tf_{ij}) + 1\right) / \left(\ln\left(l_i / nt_i\right) + 1\right)}{(1 - slope) \cdot pivot + slope \cdot nt_i}$		

Table A.1: Weighting schemes