Working Notes for the InFile Campaign : Online Document Filtering Using 1 Nearest Neighbor

Eric Gaussier, Ali Mustafa Qamar, and Vincent Bodinier

Laboratoire d'Informatique de Grenoble (LIG)
Bâtiment IMAG B - 385 avenue de la Bibliothèque
38400 Saint Martin d'Hères
{eric.gaussier,ali-mustafa.qamar,vincent.bodinier}@imag.fr

Abstract. This paper has been written as a part of the InFile (INFormation, FILtering, Evaluation) campaign. This project is a cross-language adaptive filtering evaluation campaign, sponsored by the French national research agency, and it is a pilot track of the CLEF (Cross Language Evaluation Forum) 2008 campaigns. We propose in this paper an online algorithm to learn category specific thresholds in a multiclass environment where a document can belong to more than one class. Our method uses 1 Nearest Neighbor (1NN) algorithm for classification. It uses simulated user feedback to fine tune the threshold and in turn the classification performance over time. The experiments were run on English language corpus containing 100,000 documents. The best results have a precision of 0.366 and the recall is 0.260.

Key words: filtering, classification, InFile, CLEF 2008, k Nearest Neighbor

1 Introduction

Infile campaign is aimed at testing crosslingual adaptive filtering systems. The task is to classify documents into different topics in an online fashion. In order to improve classification accuracy, a client can request for feedback. The number of feedbacks is limited to 50.

The k Nearest Neighbor (kNN) algorithm is a widely used supervised learning method and has been applied in a variety of different tasks like text classification, web-page classification etc [1–4]. It classifies a new instance based on its k closest examples in the feature space where the closeness is found using distance or similarity measures. Similarity has been preferred over distances while dealing with text. In such a case, cosine measure is used instead of Eucledian or Mahalanobis distance. The kNN rule is also refered to as a lazy method since it defers all computations to the run time. Yang et al. [1] have described a method where a category-specific threshold is learned on a validation set of examples. An example is said to be belonging to a category only if surpasses the threshold.

The rest of the paper is organized as follows. The problem is described in Section 2. Section 3 contains the proposed online algorithms followed by the experiments and results in Section 4.

2 Problem formulation

Our goal for the InFile campaign evaluation is to filter 100,000 English documents provided one by one. The filtering has to be done on 50 topics (numbered from 101 to 150). 30 of them are related with general news and events (national and international affairs, politics, sports etc.) while 20 concern scientific and technical subjects.

A document can belong to zero, one or more topics. So, the system must be able to process similarity with each topic in order to determine whether a document belongs to it or not. In this project the context of competitive intelligence was considered where information filtering is a very specific subtask of the information management process [5]. In this approach, the information filtering task is very similar to Selective Dissemination of Information (SDI), one of the original and usual function assumed by documentalists and more recently, by other information intermediaries such as technological watchers or business intelligence professionals.

3 Algorithms' description

3.1 General explanation

In this section, we present algorithms based on two different types of similarities. In the first type, a similarity between a topic file and a document is calculated (where a topic file is the profile of the topic) whereas in the other one, we compute a similarity between two documents.

It is necessary to use two similarities because topic files and documents do not have the same kind of content and hence the significance and interpretation of these two types of similarity are not the same. For instance, the similarity value between a topic file and a relevant document for this topic can be around 0.40 whereas the similarity value between two documents relevant to the same topic can be much higher. These two algorithms are based on a 1 nearest neighbor (1NN) algorithm and cosine similarity.

Each time a new document is retrieved, similarities with each of the topics are calculated. The comparison of this similarity value with a threshold determines whether the document is relevant to the topic or not.

The similarity f_i(d) between a document d and a topic i is calculated as follows :

$$f_i(\mathrm{d}) = \alpha * \cos(t_i,\mathrm{d}) + (1 - \alpha)\ max_{(d' \in t_i)} \cos(\mathrm{d},\mathrm{d'}),$$
where $\alpha \in [0,1]$

Here f_i(d) is composed of two terms. The first one is related to the similarity between a topic file and a document weighted by α. The second term represents the similarity of a document d with the nearest neighbor in the topic weighted by 1 - α (the value of the nearest neighbor similarity is equal to 1 if no document has already been assigned to the topic).

3.2 detailed algorithms

We will describe here the two algorithms developed during the course of InFile campaign. They are written in pseudo-code.

notations : t_i : topic file i $\quad (i \in \{101, 102, ..., 150\})$
d : the current document processed.

Algorithm 1 This algorithm does not use any feedback. Its principle is rather simple. A threshold called S enables us to determine whether a document d is relevant to the topic or not. Its value is calculated as follows :

S = α * β_{max} + ($1 - \alpha$) * x_s
where β_{max}, $x_s \in$ [0,1]

The threshold S is composed of two terms. The first one is β_{max} threshold weighted by α while the second one is x_s threshold weighted by $1 - \alpha$. The threshold β_{max} is the value above which we consider that the cos(t_i,d) is high enough to say that the document is relevant to the topic i. The threshold x_s is a value above which we consider that the $max_{(d' \in t_i)} cos(d, d')$ is high enough and it can be said that the document d is relevant to the topic i.

This first algorithm requires fixing of three parameters : α, x_s and $\beta_{max} \in$ [0,1].

for each new document d
  for each topic i $\quad (i \in$ {101,102,...,150})
    if ($f_i(d) \geq$ S) {Assign d to topic i}
    else {Do not assign d to topic i}

where f_i(d) = α * cos(t_i,d) + (1 - α) $max_{(d' \in t_i)}$ cos(d,d') ,
$max_{(d' \in i)} cos(d, d') = 1$ if no document is already assigned to topic i,
S = α * β_{max} + (1 - α) * x_s

Algorithm 2 The basics of the algorithm 2 are the same as that of the first one. The main difference is that two different ways are used to judge the relevance. Another difference is the threshold used for f_i(d).

The idea is to build a base of 10 documents for each topic by only using cosine with the topic file. And once this base is built, the similarity f_i(d) is used in the same way as used in the first algorithm. In this algorithm, feedbacks are used in order to limit the number of mistakes while building a base of 10 documents. γ is the threshold used to judge $cos(t_i, d)$ in the first part of the algorithm ($\gamma \in$ [0,1]).

s_i is the one used to judge the f_i(d) similarity in the second part of the algorithm. Its formula is the following :

$s_i = min_{(d \in i)}(f_i(d))$

In this algorithm two constants must be parametrized namely, α and $\gamma \in$ [0,1].

for each new document d
  for each topic i
    if (number of documents already assigned to topic $i < 10$)
      if ($cos(t_i, d) > \gamma$)
        feedback = 1
        if (number of remaining feedbacks != 0) feedback = AskFeedback()
        if (feedback == 1) {assign document d to topic i}
        else {do not assign d to topic i}
      else {do not assign d to topic i}
    else if (f_i(d) > s_i) {assign d to topic i}
        else {do not assign d to topic i}

where $f_i(d) = \alpha * cos(t_i, d) + (1 - \alpha) max_{(d' \in t_i)} cos(d, d')$,
$s_i = min_{(d \in i)}(f_i(d))$

These two algorithms' behaviours depend strongly on the relevance values of thresholds that are fixed before launching the tests (in particular, β_{max} and x_s for the first one and γ for the second one).

4 Experiments

We have used InFile English data for the experimental validation. For each new document retrieved, first of all stemming is performed using Porter's algorithm. This is followed by the removal of stop-words, XML tags skipping and the building of a document vector (which associates each term with its frequency) using rainbow [6].

During the InFile campaign evaluation, three runs have been submitted. The table below described run's features.

	Name	algorithm	parameters
Run 1	runname	1	$\alpha = 0.7$; $x_s = 0.8$; $\beta_{max} = 0.45$
Run 2	run2G	1	$\alpha = 0.7$; $x_s = 0.7$; $\beta_{max} = 0.4$
Run 3	run5G	2	$\alpha = 0.7$; $\gamma = 0.42$

Table 1. Run's features

4.1 Results

1597 is the total number of relevant documents to find during a run. The different scores have been computed by averaging scores' values on the whole profiles. The complete reports of each runs are presented in the appendix 1, 2 and 3.

Run1 :

	Relevant	Not relevant
Retrieved	152	394
Not retrieved	1445	98009

Table 2. Run 1 - results

During the first run, our system retrieved 546 documents. Among these, 152 were relevant.

Run 2 :

	Relevant	Not relevant
Retrieved	411	900
Not retrieved	1186	97503

Table 3. Run 2 - results

For the second run, 411 documents that were retrieved were relevant. Overall, 1311 documents have been retrieved during this run.

Run 3 :

	Relevant	Not relevant
Retrieved	601	7037
Not retrieved	996	91366

Table 4. Run 3 - results

7638 documents have been retrieved and 601 have been considered as correctly assigned to a topic.

Fig. 1 shows the number of relevant documents retrieved for each set of 10000 documents. Fig. 2 depicts the different measures evolution during the run 3.

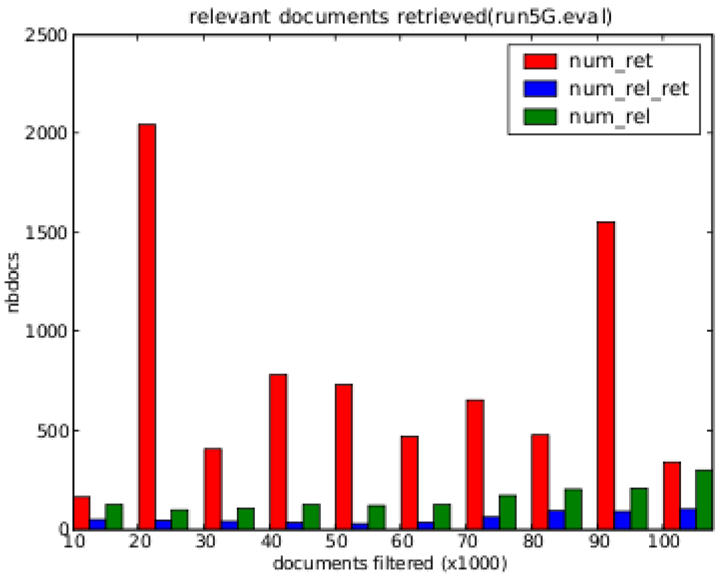


Fig. 1. run5G - Relevant Documents Retrieved

Abbreviations : num_ret for 'number of documents retrieved', num_rel_ret for 'number of relevant documents retrieved' and num_rel for 'number of relevant documents'.

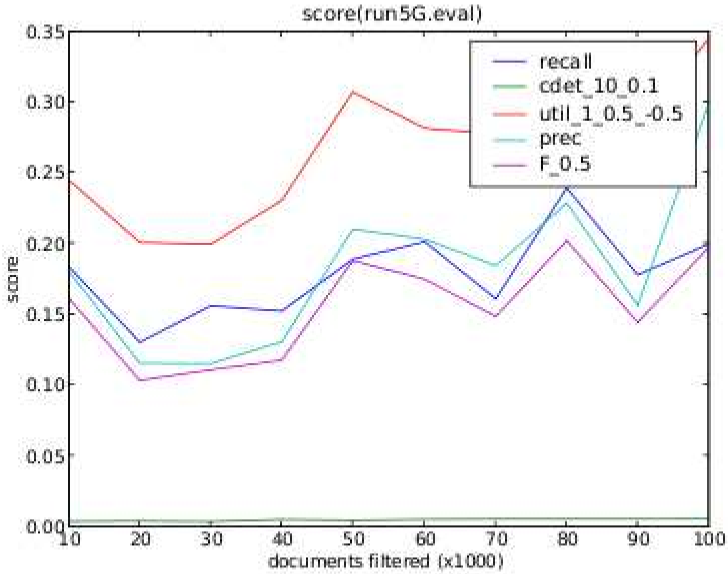


Fig. 2. run5G - Scores' Evolution

Abbreviations : prec for 'precision', F_0.5 for 'F-measure', util_1_0.5_-0.5 for 'linear utility' and cdet_10_0.1 for 'detection cost'.

Average scores : Here is a table which contains the whole average scores.

	Precision	Recall	F-measure	Linear Utility	Detection Cost
Run 1	0.366	0.068	0.086	0.311	0.009
Run 2	0.357	0.165	0.165	0.335	0.008
Run 3	0.306	0.260	0.209	0.351	0.007

Table 5. runs scores

4.2 Analysis

Run 1 scores are rather low. In particular, recall's value is very low whereas precision is around 0.36 which shows that this run is precision-oriented because it retrieves a tiny part of the whole documents. Run 2 precision value is close to the first run. The recall and the F-measure are slightly better. This run is also precision-oriented with a precision's value clearly better than the recall one.

If we consider the overall scores, run 3 is better than the two others. Although the precision is slightly lower, the recall's score attains 0.26 while the F-measure reaches 0.2.

The overall detection cost is very low during the runs (less than 0.01). This a strong point for our algorithms. We can also notice that the linear utility progressively increases between 0.2 and 0.3.

The run 1 is a more conservative method compared to the run 2 because of the differences in β_{max} and x_s values which affect the value of S. Hence the run 2 is expected to assign more documents to the topics than the run 1.

Regarding measures evolution, for run 1, precision, recall and F-measure tend to decrease slightly. For the run 2, they randomly vary but remain the same at the end, whereas they increase slightly during the last run.

Basically, these runs are clearly precision-oriented. Indeed, for the three runs respectively 0.5, 1.3 and 7.6 percent of the number of documents are retrieved.

5 Conclusion

Our participation at this project was a good experience and it enables us to take awareness of the specificity of the information filtering in the frame of competitive intelligence. The major difficulty is to design a system with very quick adaptivity because of the few feedbacks available. Indeed, the system must learn metrics on few data.

Since we were the only participants who have completed the task during the InFile project, it is difficult to appreciate the results. Comparison with other systems would enable us to have a better analysis of our results.

As a consequence, we cannot say that our system is good but it clearly seems that the scores obtained are not sufficient to fulfill this task. Indeed, we cannot say that the better F-score we have obtained (around 0.2) is a sufficient one for this specific task which requires much higher scores.

The interest in these experiments actually remains in the way we computed similarity and judged its relevance. Algorithms presented in this paper are characterized by their obvious simplicity and effectiveness. We could imagine more complex algorithms based on k Nearest Neighbor with k greater than one and an attempt to learn metric using feedbacks. However, it is not expected to give better results. Indeed, we do not think that the use of a higher k would give different results since the topics are relatively distant from each other. In general, a document is considered at the most close to one topic, so there is no conflict at this level. Moreover, learning a metric with only 50 feedbacks is difficult.

Actually, the difficulty lies in calculating the similarities and finding the decision bounds. In order to refine the results, a solution could be to attach more importance to the content documents processing (for instance, by working on n-grams rather than on stems) in order to be more precise.

References

1. Yiming Yang and Xin Liu. A re-examination of text categorization methods. In *SIGIR '99: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 42–49, New York, NY, USA, 1999. ACM Press.
2. Li Baoli, Lu Qin, and Yu Shiwen. An adaptive k-nearest neighbor text categorization strategy. *ACM Transactions on Asian Language Information Processing (TALIP)*, 3(4):215–226, 2004.
3. Oh-Woog Kwon and Jong-Hyeok Lee. Web page classification based on k-nearest neighbor approach. In *IRAL '00: Proceedings of the fifth international workshop on on Information retrieval with Asian languages*, pages 9–15, New York, NY, USA, 2000. ACM Press.
4. Fabrizio Sebastiani. Machine learning in automated text categorization. *ACM Comput. Surv.*, 34(1):1–47, 2002.
5. Romaric Besancon, Stéphane Chaudiron, Djamel Mostefa, Ismail Timimi, and Khalid Choukri. The infile project: a crosslingual filtering systems evaluation campaign. In European Language Resources Association (ELRA), editor, *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco, may 2008.
6. A. K. McCallum. Bow: A toolkit for statistical language modeling, text retrieval, classification and clustering, 1996.

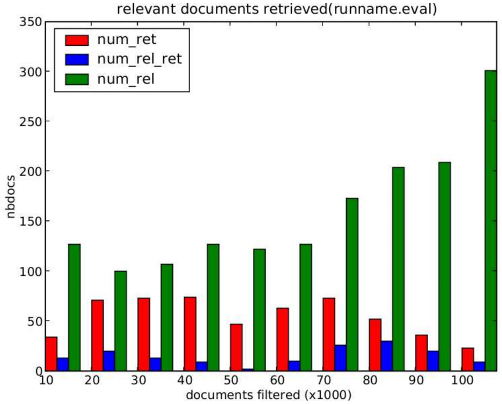
APPENDIX 1 : run 1 complete report

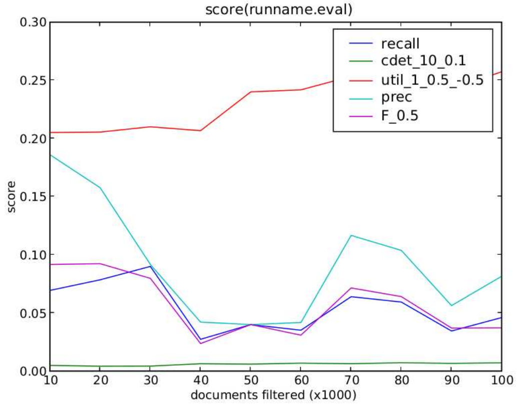
team:	IMAG
runName:	runname
source language:	eng
target language:	eng
topic fields:	all

overall statistics(runname.eval)
total number of documents for all queries

Retrieved	546
Relevant	1597
Relevant retrieved	152

Adaptation Results (according to the number of documents filtered)





All scores by query

runname.eval	prec	recall	F_0.5	util_1_0.5_-0.5	cdet_10_0.1
101	0.800	0.048	0.091	0.361	0.010
102	1.000	0.087	0.160	0.391	0.009
103	1.000	0.067	0.125	0.378	0.009
104	0.000	0.000	0.000	0.238	0.010
105	0.000	0.000	0.000	0.318	0.010
106	0.000	0.000	0.000	0.319	0.010
107	0.000	0.000	0.000	0.321	0.010
108	0.000	0.000	0.000	0.000	0.010
109	1.000	0.036	0.069	0.357	0.010
110	0.000	0.000	0.000	0.000	0.000
111	0.000	0.000	0.000	0.222	0.010
112	0.000	0.000	0.000	0.333	0.010
113	1.000	0.043	0.083	0.362	0.010
114	0.000	0.000	0.000	0.211	0.010
115	0.000	0.000	0.000	0.250	0.010
116	0.000	0.000	0.000	0.000	0.010
117	0.000	0.000	0.000	0.314	0.010
118	0.120	0.608	0.201	0.000	0.004
119	0.000	0.000	0.000	0.333	0.010
120	1.000	0.250	0.400	0.500	0.007
121	0.000	0.000	0.000	0.333	0.010
122	0.000	0.000	0.000	0.333	0.010
123	0.000	0.000	0.000	0.333	0.010
124	0.000	0.000	0.000	0.325	0.010
125	0.923	0.414	0.571	0.598	0.006
126	0.000	0.000	0.000	0.327	0.010
127	0.500	0.010	0.019	0.337	0.010
128	0.000	0.000	0.000	0.329	0.010
129	1.000	0.016	0.031	0.344	0.010
130	1.000	0.067	0.125	0.378	0.009
131	0.000	0.000	0.000	0.333	0.010
132	1.000	0.231	0.375	0.487	0.008
133	0.000	0.000	0.000	0.167	0.010
134	0.000	0.000	0.000	0.000	0.000
135	1.000	0.067	0.125	0.378	0.009
136	1.000	0.050	0.095	0.367	0.009
137	0.000	0.000	0.000	0.330	0.010
138	0.000	0.000	0.000	0.315	0.010
139	1.000	0.020	0.039	0.347	0.010
140	0.000	0.000	0.000	0.167	0.010
141	0.000	0.000	0.000	0.327	0.010
142	1.000	0.021	0.042	0.348	0.010
143	0.867	0.243	0.380	0.483	0.008
144	0.000	0.000	0.000	0.333	0.010
145	1.000	0.035	0.068	0.357	0.010
146	0.727	0.667	0.696	0.694	0.003
147	0.000	0.000	0.000	0.000	0.000
148	0.857	0.409	0.554	0.583	0.006
149	0.500	0.031	0.059	0.344	0.010
150	0.000	0.000	0.000	0.325	0.010
all	0.366	0.068	0.086	0.311	0.009

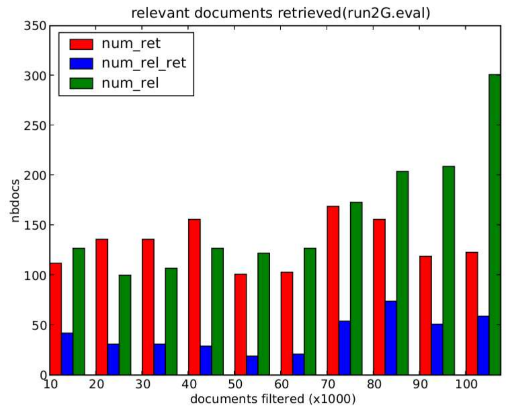
APPENDIX 2 : run 2 complete report

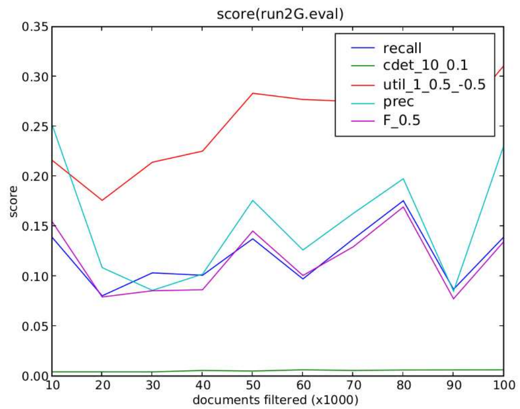
team:	IMAG
runName:	run2G
source language:	eng
target language:	eng
topic fields:	all

overall statistics(run2G.eval)
total number of documents for all queries

Retrieved	1311
Relevant	1597
Relevant retrieved	411

Adaptation Results (according to the number of documents filtered)





All scores by query

run2G.eval	prec	recall	F_0.5	util_1_0.5_-0.5	cdet_10_0.1
101	0.857	0.072	0.133	0.378	0.009
102	0.000	0.000	0.000	0.319	0.010
103	0.000	0.000	0.000	0.311	0.010
104	0.000	0.000	0.000	0.286	0.010
105	1.000	0.045	0.087	0.364	0.010
106	0.000	0.000	0.000	0.319	0.010
107	0.741	0.714	0.727	0.726	0.003
108	0.000	0.000	0.000	0.000	0.010
109	0.000	0.000	0.000	0.321	0.010
110	0.000	0.000	0.000	0.000	0.000
111	0.000	0.000	0.000	0.222	0.010
112	0.000	0.000	0.000	0.333	0.010
113	0.000	0.000	0.000	0.319	0.010
114	0.000	0.000	0.000	0.000	0.010
115	0.000	0.000	0.000	0.250	0.010
116	0.000	0.000	0.000	0.000	0.010
117	0.000	0.000	0.000	0.314	0.010
118	0.093	0.861	0.167	0.000	0.002
119	0.000	0.000	0.000	0.333	0.010
120	0.273	0.750	0.400	0.167	0.003
121	0.000	0.000	0.000	0.325	0.010
122	0.000	0.000	0.000	0.278	0.010
123	0.000	0.000	0.000	0.333	0.010
124	0.486	0.447	0.466	0.474	0.006
125	0.854	0.707	0.774	0.764	0.003
126	0.871	0.482	0.621	0.631	0.005
127	0.690	0.467	0.557	0.575	0.005
128	0.412	0.197	0.267	0.371	0.008
129	0.333	0.016	0.030	0.333	0.010
130	0.900	0.300	0.450	0.522	0.007
131	1.000	0.071	0.133	0.381	0.009
132	1.000	0.231	0.375	0.487	0.008
133	0.000	0.000	0.000	0.167	0.010
134	0.000	0.000	0.000	0.000	0.000
135	1.000	0.067	0.125	0.378	0.009
136	1.000	0.050	0.095	0.367	0.009
137	0.800	0.045	0.086	0.360	0.010
138	0.000	0.000	0.000	0.315	0.010
139	0.000	0.000	0.000	0.327	0.010
140	0.000	0.000	0.000	0.167	0.010
141	0.000	0.000	0.000	0.327	0.010
142	1.000	0.021	0.042	0.348	0.010
143	0.511	0.664	0.577	0.564	0.003
144	0.000	0.000	0.000	0.333	0.010
145	1.000	0.018	0.034	0.345	0.010
146	0.500	0.750	0.600	0.583	0.003
147	0.000	0.000	0.000	0.000	0.000
148	0.766	0.818	0.791	0.795	0.002
149	1.000	0.188	0.316	0.458	0.008
150	0.769	0.256	0.385	0.479	0.007
all	0.357	0.165	0.165	0.335	0.008

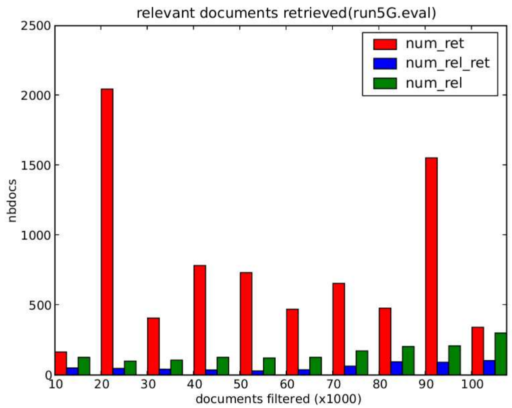
APPENDIX 3 : run 3 complete report

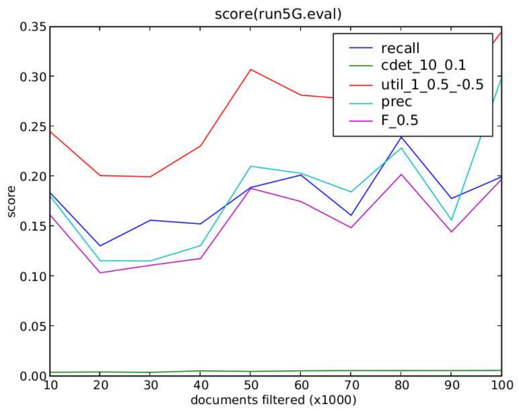
team:	IMAG
runName:	run5G
source language:	eng
target language:	eng
topic fields:	all

overall statistics(run5G.eval)
total number of documents for all queries

Retrieved	7638
Relevant	1597
Relevant retrieved	601

Adaptation Results (according to the number of documents filtered)





All scores by query

run5G.eval	prec	recall	F_0.5	util_1_0.5_-0.5	cdet_10_0.1
101	0.840	0.253	0.389	0.486	0.007
102	0.235	0.174	0.200	0.261	0.008
103	0.250	0.067	0.105	0.311	0.009
104	0.048	0.143	0.071	0.000	0.009
105	0.000	0.000	0.000	0.333	0.010
106	0.000	0.000	0.000	0.250	0.010
107	0.014	0.857	0.028	0.000	0.003
108	0.000	0.000	0.000	0.333	0.010
109	0.500	0.036	0.067	0.345	0.010
110	0.000	0.000	0.000	0.000	0.000
111	0.000	0.000	0.000	0.333	0.010
112	0.000	0.000	0.000	0.333	0.010
113	0.000	0.000	0.000	0.333	0.010
114	0.000	0.000	0.000	0.158	0.010
115	0.053	0.250	0.087	0.000	0.008
116	0.500	1.000	0.667	0.667	0.000
117	0.500	0.118	0.190	0.373	0.009
118	0.016	0.962	0.031	0.000	0.005
119	0.000	0.000	0.000	0.333	0.010
120	0.125	0.750	0.214	0.000	0.003
121	0.000	0.000	0.000	0.333	0.010
122	0.000	0.000	0.000	0.333	0.010
123	0.000	0.000	0.000	0.333	0.010
124	0.479	0.605	0.535	0.518	0.004
125	0.875	0.845	0.860	0.856	0.002
126	0.607	0.661	0.632	0.631	0.003
127	0.516	0.619	0.563	0.552	0.004
128	0.436	0.239	0.309	0.390	0.008
129	0.000	0.000	0.000	0.333	0.010
130	0.818	0.300	0.439	0.511	0.007
131	0.000	0.000	0.000	0.333	0.010
132	1.000	0.462	0.632	0.641	0.005
133	0.000	0.000	0.000	0.333	0.010
134	0.000	0.000	0.000	0.000	0.000
135	1.000	0.200	0.333	0.467	0.008
136	0.875	0.350	0.500	0.550	0.007
137	0.624	0.602	0.613	0.614	0.004
138	0.000	0.000	0.000	0.333	0.010
139	0.000	0.000	0.000	0.333	0.010
140	0.000	0.000	0.000	0.333	0.010
141	1.000	0.020	0.040	0.347	0.010
142	0.000	0.000	0.000	0.333	0.010
143	0.245	0.860	0.382	0.025	0.002
144	0.000	0.000	0.000	0.333	0.010
145	0.800	0.070	0.129	0.374	0.009
146	0.500	0.917	0.647	0.639	0.001
147	0.000	0.000	0.000	0.000	0.000
148	0.724	0.955	0.824	0.848	0.000
149	1.000	0.312	0.476	0.542	0.007
150	0.737	0.359	0.483	0.530	0.006
all	0.306	0.260	0.209	0.351	0.007