Novelty as a Form of Contextual Re-ranking: Efficient KLD Models and Mixture Models

Ronald T. Fernández
Grupo de Sistemas Inteligentes
Departamento de Electrónica y Computación
Universidad de Santiago de Compostela, Spain
ronald.teijeira@rai.usc.es

David E. Losada
Grupo de Sistemas Inteligentes
Departamento de Electrónica y Computación
Universidad de Santiago de Compostela, Spain
dlosada@usc.es

ABSTRACT

Current Information Retrieval systems are often based on topicality. They estimate relevance by comparing the similarity between the user query and each document. These systems do not take into account important contextual information. More specifically, they do not often apply mechanisms to filter out redundant information. We interpret context here as the set of chunks of text from the ranked set of documents that the user has already seen. This is a valuable contextual information to guide the retrieval processes in a way that avoids redundancy. It is desirable that the ranking of results is composed by relevant but also novel material. This means that each document must provide to the user unseen information which is related to his need.

In this work we study different novelty detection approaches that make good use of this contextual information. We show that these techniques can be applied effectively and efficiently at the sentence level.

Symposium Themes: Context-aware retrieval models.

1. INTRODUCTION

Information Retrieval (IR) systems are often based exclusively on topicality. Topicality involves some form of matching between the query terms and the document terms. However, this ignores important contextual information that could be used to enhance users's satisfaction. The pieces of text that the user has already seen (e.g. having read a number of paragraphs from documents in the ranks *1* to *k–1* before reading the document in the top k position) is a valuable source of information that can be used to avoid redundancy. Redundant information is not usually desirable. The reason is that users are often more concerned about looking for new information (novel documents) and they are less tolerant of getting information they already know [7].

We can apply some novelty detection mechanism to filter out redundant information and, therefore, alleviate this problem. IR systems which employ some kind of novelty detection provide relevant and non-redundant results to the users and, so, it will be easier and faster for them to find the information that actually satisfies their needs.

IIiX 2008 London, UK

To study these issues properly, we adopt the novelty task as defined by the TREC novelty tracks [7, 12, 11]. This simulates a retrieval process in which a user walks down a ranked list of documents and views some sentences from the retrieved documents. The aim is to present the user with key sentences that are novel with respect to the sentences seen before. The history of seen sentences is therefore a form of context that summarises the information that the user has acquired within the retrieval process so far. Modelling this context will help IR systems to detect redundancy and show only novel information.

The novelty detection task starts form a list of sentences ranked using topicality. It is assumed that the user reviews this list following the order induced by topicality (as a matter of fact, the assessors that created the novelty judgements were required to follow this order) and finds sentences which are either novel or redundant. The objective is to design automatic ways to re-rank the ranking of sentences to promote novelty (i.e. redundant sentences should be pushed down in the final ranking).

We can find in the literature different approaches which address this problem, such as New Words, Set Difference and Cosine Distance [1]. However, these methods are very simplistic. For instance, New Words is based on counting the number of new terms (not seen before) that a sentence contains.

We study here two approaches based on modelling the context of seen sentences in a formal way. The first one is an effective approach based on Language Modelling that addresses the problem in a very efficient way (we consider that efficiency and effectiveness are both very important in this sort of applications). In particular, we experiment with a variation of a Kullback-Leibler Divergence (KLD) model whose computational cost is significantly lower than the cost of the original KLD model [1]. The second technique that we study here is an alternative method that has the advantage of being independent of any parameter tuning. Most novelty detection techniques are usually dependent on some internal parameter values. Since performance is often strongly dependent on parameter tuning, it is desirable to develop effective methods to estimate the parameters automatically. To this aim, we apply the Expectation-Maximisation (EM) algorithm to learn the parameters of a Mixture Model that combines the model of each sentence with the model of the previous sentences. Both methods are tested against standard benchmarks and compared with popular techniques showing that these methods are competitive.

The rest of the paper is organised as follows. Section 2 reviews some previous studies related to our research. In section 3 we explain the foundations of our contribution and we evaluate empirically the results in section 4. The paper ends with some conclusions.

2. RELATED WORK

Many novelty detection approaches have been proposed in the past. One of the seminal studies in this subject is based on Maximum Marginal Relevance (MMR) [3]. This study combines linearly an estimation of relevance (query-document similarity score) with an estimation of redundancy of the document with respect to the documents ranked above. In the context of Information Filtering [16] some novelty measures were proposed and evaluated. More specifically, a cosine similarity measure and a redundancy measure based on a mixture of three Language Models were proposed. However, this sort of Mixture Models have not been tested for novelty detection at sentence level. This is one of the objectives of our work.

The approaches sketched above estimate novelty at document level. We can also find in the literature several methods to address the novelty detection problem at sentence level. The simplest ones are New Words, Set Difference and Cosine Distance [1]. These techniques are based on some form of matching between each sentence and the previous ones in the ranking of sentences. These approaches have been proved to work reasonably well, but they lack a formal modelling of the elements involved.

Language Models (LMs) are powerful tools that proved to work well in IR [4]. Since the seminal proposals in the late nineties [10], many other studies have proposed LMs in a number of IR problems. In particular, LMs have received some attention in the context of sentence retrieval and novelty detection. For instance, in [1] a LM is created for each current sentence and another LM is created for set of previously seen sentences. The authors propose to obtain novelty scores by applying the divergence between both models. Additionally, in [5], the authors used this model to study the impact of smoothing on novelty detection performance.

Other researchers have approached the problem from a different perspective. In [6], Local Context Analysis [13] was used to define a query-oriented vocabulary that was applied to drive novelty detection. This helps to avoid redundant sentences and it is particularly useful as a high precision mechanism.

In [15] the authors evaluated a Mixture Model that incorporates novelty detection for subtopic retrieval. The Mixture Model's parameter was estimated automatically and this estimation yielded to good performance in terms of subtopic coverage. We adopted this approach in our work to test its performance for novelty detection at sentence level.

3. NOVELTY DETECTION AT SENTENCE LEVEL

Simple novelty detection methods have been defined in the literature: *New Words*, *Set Difference* and *Cosine Distance* [1]. In *New Words*, the novelty score of a sentence is defined as the number of sentence terms that do not appear in any previous sentence. In *Set Difference*, however, the novelty score for a sentence is determined by the difference between the sentence and its most similar sentence in the history of seen sentences (the sentence-to-sentence difference is computed as a set difference between the respective set of terms in the sentence). The higher similarity between a sentence and its most similar sentence, the lower the novelty score assigned. With *Cosine Distance*, the novelty score for a sentence is computed in a similar way as *Set Difference* does, but the sentence-to-sentence similarity is computed using the cosine measure with the usual representation of sentences as vectors.

We propose here two different methods to address this problem using the power and robustness of Language Models. On one hand, we propose an efficient variation of the so-called Non-Aggregate Model [1]. On the other hand, we apply Mixture Models for novelty detection and check whether or not successful models applied for subtopic retrieval are usable here. These models are discussed in the next subsections.

3.1 Non-Aggregate Model

The *Aggregate Model* (AM) and *Non-Aggregate Model* (NAM) [1] are two alternative formal methods based on Language Modelling [4]. Both of them apply KLD to compute novelty. KLD measures the divergence between two probability distributions (p_1 and p_2) as follows:

$$KLD(p_1||p_2) = \sum_x p(x|p_1) \log \frac{p(x|p_1)}{p(x|p_2)} \quad (1)$$

In the context of IR, given two LMs associated to documents (or sentences) d_i and d_j, the expression above can be rewritten as:

$$KLD(d_i||d_j) = \sum_t p(t|d_i) \log \frac{p(t|d_i)}{p(t|d_j)} \quad (2)$$

where the sum goes on every term t in the vocabulary.

Given a set of sentences ranked by estimated relevance, NAM generates an individual LM for each sentence and, next, computes the KLD between the LM of the current sentence and the LM of every sentence ranked above. The novelty score is obtained as the minimum value obtained across these pairwise operations. Formally,

$$N(s_i) = \min(KLD(s_i||s_1), ..., KLD(s_i||s_{i-1}))$$

where s_i is the LM for the current sentence and $s_j, j = 1...i-1$ are the LMs of each one of the previously seen sentences.

In contrast, AM generates a LM for the current sentence and another one for all the previous sentences. This means that the contextual information is treated as a single unit that represents the user's interaction within the retrieved set of documents. The score is simply the KLD between the LM of the current sentence and the LM of the set of previously seen sentences.

Because NAM provides better results for novelty detection at sentence level than AM [5], we will focus here on the NAM approach.

To generate the LMs, we apply Dirichlet smoothing (as indicated in [5], Dirichlet smoothing works better than Jelinek-Mercer). Sentence LMs need to be properly smoothed in order to achieve good performance in novelty detection [5]. Smoothing makes that any probability in equation 2 is greater than zero and, consequently, we need to traverse all the vocabulary terms to compute KLD. Note also that NAM requires many KLD operations for computing the novelty score of each sentence. It is therefore very important to reduce the computational cost associated to every individual sentence-to-sentence novelty score computation. This motivation stands behind the variation proposed next.

3.1.1 NAM-Quick: Efficient Non-Aggregate Model

NAM is computationally inefficient because it requires to go across all the vocabulary terms multiple times to compute the novelty score for each sentence. This problem gets worse as we compute the novelty score for sentences down in the list that require revisiting many previous sentences for checking redundancy.

We propose a method that tries to alleviate this problem. This new technique will be referred to as *NAM-Quick* and it is a simple variation of *NAM* that computes an approximation of the KLD values. Instead of traversing the whole vocabulary to compute KLD, *NAM-Quick* considers only the subset of terms which belong to at least one of the sentences involved. Formally,

$$KLD^{*}(s_i||s_j) = \sum_{t \in s_i \vee s_j} P(t|s_i) \log \frac{P(t|s_i)}{P(t|s_j)}$$

We use the notation KLD^{*} to emphasise that this is an approximation to the real KLD value. The novelty score is computed as the pairwise operations using this version of KLD:

$$N(s_i) = \min(KLD^{*}(s_i||s_1), ..., KLD^{*}(s_i||s_{i-1}))$$

We expect that this is not only a more efficient method but also provides better performance results than *NAM*. Terms that are not mentioned by any of the sentences involved might introduce some noise in the computation of novelty. The major contribution to the KLD score comes from the terms that appear in at least one of the sentences (e.g. a term appearing in s_i and missing in s_j). These terms might boost novelty because they usually have low probability mass into one LM (s_j) and high probability in the other LM (s_i). On the other hand, terms that are missing in both sentences have usually marginal probability values assigned and, therefore, their contribution to the novelty score is very low.

3.2 Mixture Model

In [15], a Mixture Model approach was applied to model the degree of redundancy of chunks of text or documents in a subtopic detection problem.We check here whether this method is usable in the context of novelty detection at sentence level as defined in the TREC novelty tracks [7, 12, 11]. The main advantage of this method is that it estimates automatically the parameters involved. Unlike the NAM techniques (which require parameter tuning to smooth the LMs properly [5]) the Mixture Model, as defined in [15], can adjust its internal parameter in an automatic way. We now describe the formulation of this model adapted to the detection of novel sentences.

The generation of sentences within the documents is regarded as a random process where two LMs are involved: a background LM and a reference LM. In subtopic retrieval, the background LM models the general use of the language while the reference LM models the information of documents already seen. The more a sentence is *explained* by the reference model, the less novel the sentence is. In contrast, if the sentence deviates significantly from the reference model then it is likely novel. Formally, the novelty detection method (adapted to work at sentence level) is based on the log likelihood as follows:

$$l(\lambda|s_i, \theta_{R_{s_j}}) = \sum_{j=1}^{n_{s_i}} \log((1-\lambda)p(t_j|\theta_{R_{s_j}}) + \lambda p(t_j|\theta_B)) \quad (3)$$

where s_i is the current sentence, n_{s_i} is the number of terms in sentence s_i, $\theta_{R_{s_j}}$ is the reference model (it models a sentence s_j from the history of sentences), and θ_B is the background model (it models the use of terms in a large collection).

This expression models the likelihood of the sentences under the interpretation that sentences are generated by the models θ_{Rs_j} and θ_B. If we now apply the EM-algorithm to obtain the maximum likelihood estimate for λ, i.e. $N(s_i, s_j) = arg\ max_{\lambda} l(\lambda|s_i, \theta_{R_{s_j}})$, the value of λ obtained can be regarded as an estimation of novelty. This is because λ is the mixing weight associated to θ_B and $(1-\lambda)$ is the mixing weight associated to $\theta_{R_{s_j}}$. The higher λ, the less connection between the current sentence and the sentence seen before ($\theta_{R_{s_j}}$).

When we are computing the novelty score for sentence s_i, we need to obtain the value of λ considering all the $i-1$ previous sentences. In the expression above we only considered two sentences s_i and s_j and, therefore, we need to repeat this process $i-1$ times. Our purpose is to get the value of λ that maximises expression 3 for every $\theta_{R_{s_j}}$ model ($j = 1...i-1$). To get the final novelty score for s_i we compute the minimum of all these $i-1$ values:

$$N(s_i) = \min(N(s_i, s_1), ..., N(s_i, s_{i-1}))$$

The EM algorithm is based on two steps: *E-Step* and *M-step*. The first step obtains the expected value for the complete log-likelihood with respect to the unknown data given the observed data. In our case, the unknown data is simply the information about which distribution ($\theta_{R_{s_j}}$ or θ_B) generated which observation (term). This can be encoded as n_{s_i} variables (y_k) that indicate whether term k was extracted from the first model ($\theta_{R_{s_j}}$) of from the second model (θ_B). That is, $y_k = 1$ means that term i was generated from $\theta_{R_{s_j}}$ and $y_k = 2$ means that i was generated from θ_B. The *M-step* maximises the expectation we computed in the previous step.

Let us consider that $\alpha_1 = (1-\lambda)$ and $\alpha_2 = \lambda$. These values are initialised to a given value (e.g. 0.9 and 0.1) and the algorithm updates these values after each iteration. The *E-Step* consists of computing the probability that each term t_k of the sentence s_i has been extracted from each model ($\theta_{R_{s_j}}$ or θ_B), given the current parameter configuration $\Theta = \{\alpha_1, \alpha_2\}$. Formally,

$$p(y_i = 1|t_k, \Theta) = \frac{\alpha_1 p(t_k|\theta_{R_{s_j}})}{\alpha_1 p(t_k|\theta_{R_{s_j}}) + \alpha_2 p(t_k|\theta_B)}$$

$$p(y_i = 2|t_k, \Theta) = \frac{\alpha_2 p(t_k|\theta_B)}{\alpha_1 p(t_k|\theta_{R_{s_j}}) + \alpha_2 p(t_k|\theta_B)}$$

The second step (*M-Step*) computes α_1 and α_2 following the expressions:

$$\alpha_1 = \frac{1}{n} \sum_{k=1}^{n_{s_i}} p(y_i = 1|t_k, \Theta)$$

$$\alpha_2 = \frac{1}{n} \sum_{k=1}^{n_{s_i}} p(y_i = 2|t_k, \Theta)$$

The algorithm has the property of converging always to a local maximum of the likelihood function.

We show in the next section the results that we obtained with this approach to estimate automatically the parameters and we compare them with other novelty approaches.

4. EXPERIMENTS

We have employed the three data collections which were made available in the context of the TREC Novelty tracks in 2002, 2003 and 2004 [7, 12, 11]. These are one of the few novelty detection benchmarks available. In 2002 and 2003 the ranking of documents given a query provided by NIST consisted only of relevant documents. In 2004, the data track is more realistic because the ranked set of documents contains both relevant and non-relevant documents. Given these rankings of documents, the task consists of selecting sentences relevant to the query that are non-redundant (i.e. supply information not covered by previously selected sentences).

To study novelty we first generated a ranking of estimated relevant sentences using a variation of tf-idf. More specifically, we used a metric which has proved to be very effective in the past [1]:

$$tf\text{-}isf(s, q) = \sum_{t \in q} \log(tf_{t,q} + 1) \log(tf_{t,s} + 1) \log \frac{n + 1}{0.5 + sf_t}$$

where $tf_{t,q}$ is the number of times term t occurs in the query q, $tf_{t,s}$ indicates the number of terms the term t occurs in the sentence s, sf_t is the number of sentences in which term t appears, and n is the number of sentences in the collection.

The top ranked sentences were next re-ordered as follows. Sentences are considered in the same order in which the documents were originally ranked by NIST and multiple sentences from the same document are considered in the order in which they appear in the document. This method has already been applied in the past [1, 5]. As matter of fact, this re-ordering was also applied by NIST before presenting the sentences to the assessors for judging novelty [7, 12, 11]. This ranked list of sentences (which will be made referred to as BDOC) can be regarded as a baseline that simply consists of taking the presumed relevant sentences in the order in which they occur in the ranked list of documents. We also experimented with other alternatives, such as BNN (tested in [8]), which uses directly the rank of relevant sentences with no re-ranking but it is clearly inferior to BDOC. Anyway, a thorough comparison among different novelty baselines is out of the scope of this work. Our purpose is to check whether the two methods explained above are usable and perform at least as well as some novelty techniques applied in the past.

Our experiments considered two types of query: *short* and *long*. *Short queries* are those queries composed by a few keywords extracted from the title tag of the TREC topic. *Long queries* are composed by terms coming from the title, description and narrative fields. Stopwords were removed in our experiments but stemming was not applied.

To compare performance, we used P@10 and P@30 because precision at top ranks are meaningful in real applications where users only want to go through a small number of sentences [9].

4.1 NAM-Quick Results

The first aim of this evaluation is to compare NAM and NAM-Quick and, in particular, determine whether or not NAM-Quick can perform equivalently to NAM. In Figure 1 and Figure 2 we show the results of P@10 and P@30, respectively, for short and long queries against the three TREC benchmarks.

The behaviour of the NAM model with varying levels of smoothing was thoroughly studied in [5]. We focus here on the relative merits of NAM-Quick against NAM. In terms of P@10, NAM-Quick provides better results than NAM across most smoothing levels. With P@30, NAM and NAM-Quick perform roughly the same. The optimal performance tends to be located at similar P@30 levels. For non-optimal smoothing levels, there is not a clear advantage of one model over the other.

NAM-Quick is not only simpler but also yields better P@10 than NAM for detection of novel sentences. This is evidence to support the claim that terms missing in the sentences to be scored are noisy for novelty detection. If the aim is to select 10 relevant and novel sentences then it is advisable to focus the search on the terms explicitly mentioned rather than considering the vocabulary as a whole. This means that the *context* under which the novelty of a sentence is assessed should be local rather than global.

Additionally, NAM-Quick is faster than NAM. This is due to the fact that NAM-Quick does not traverse through every term in the vocabulary to compute KLD. We conducted time measurements to quantify this improvement. The time savings are considerably important, as shown Table 1.

The time to process each query is significantly lower with NAM-Quick for both types of queries. The results obtained are promising. NAM was proved to work well in the past [1, 5] but it is computationally expensive. However, the simple variation designed here is not only much faster but also yields better performance.

4.2 Mixture Models

NAM and NAM-Quick's performance is dependent on the parameter that adjusts Dirichlet's smoothing in the LMs of the sentences involved. The Mixture Model explained in section 3.2 can estimate automatically its internal parameter using the EM-algorithm. This novelty detection approach was not tested before against these benchmarks. It is there-

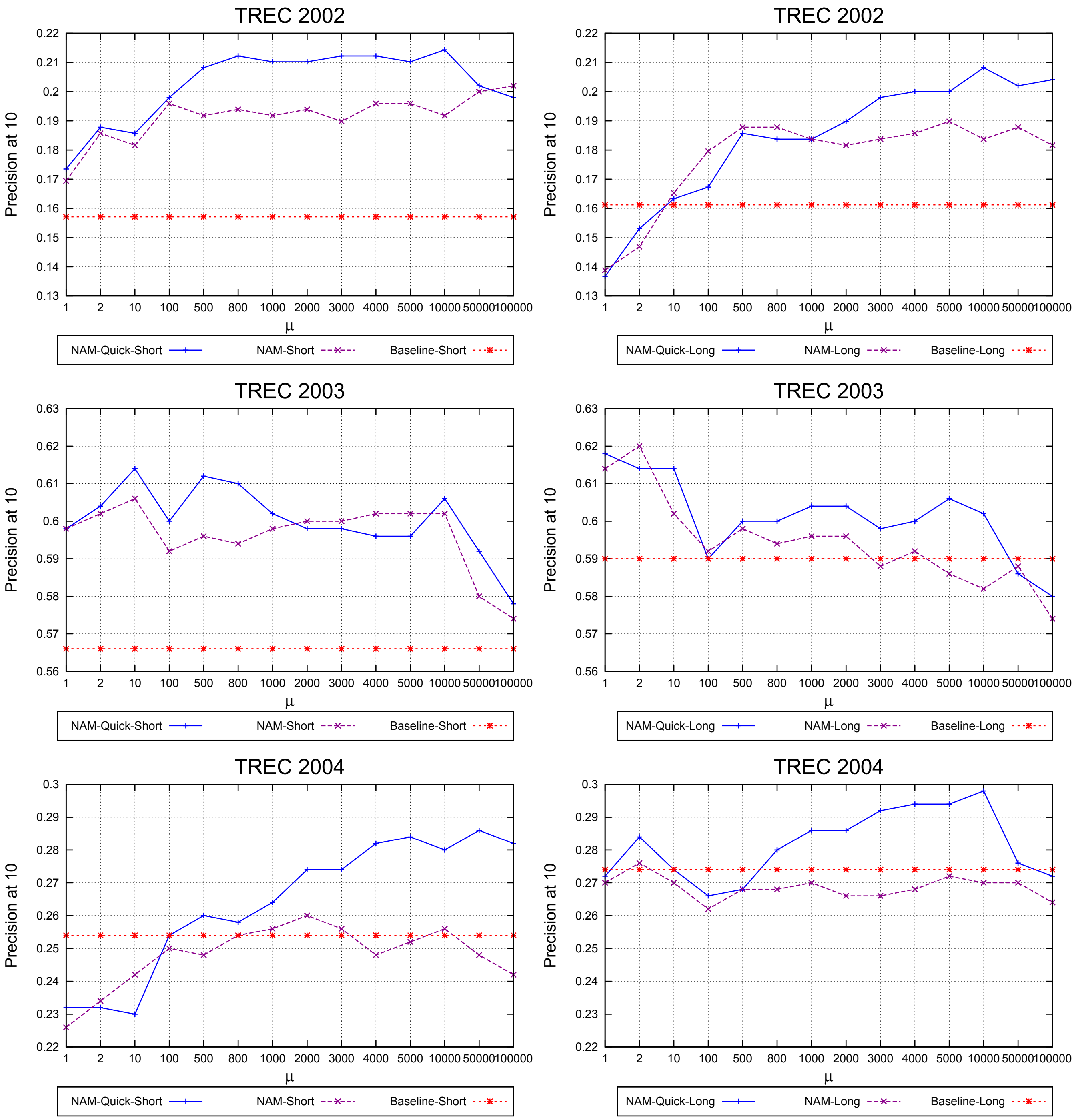

Figure 1: P@10 results for short and long queries.

	TREC 2002		TREC 2003		TREC 2004	
	NAM-Quick	*NAM*	*NAM-Quick*	*NAM*	*NAM-Quick*	*NAM*
Short Q.	0.8356	69.5734	0.4843	34.7803	1.6955	81.4953
Long Q.	2.2935	156.4768	0.9421	36.3784	1.8024	82.9485

Table 1: Time per query (seconds).

fore unclear whether this parameter estimation using EM is competitive. To check this we compare the Mixture Model results with the results obtained with NAM and NAM-Quick approaches.

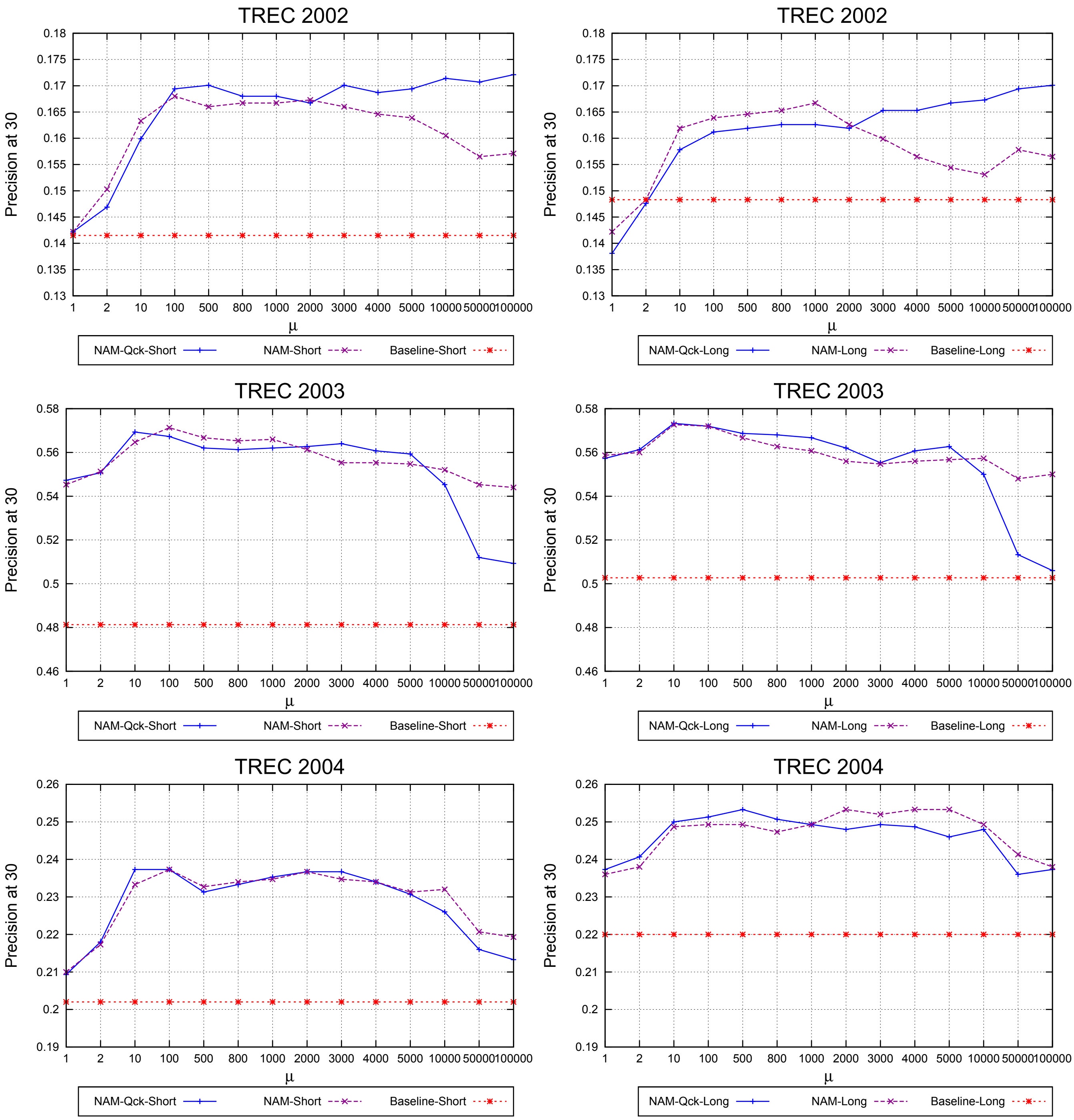

Figure 2: P@30 results for short and long queries.

Tables 2 and 3 report P@10 and P@30 for the Mixture Model approach. We compare them with the best results we obtained employing the NAM and NAM-Quick methods. The P@10 and P@30 obtained with NAM and NAM-Quick averaged across all smoothing levels are also included in the tables. The difference between the baseline and each novelty detection method has been tested for statistical significance using the t-test (95% confidence level). The novelty techniques whose score is marked with an asterisk have a precision that is statistically different from the baseline's precision. The results of the Mixture Model are rather disappointing. The Mixture Model gets usually performance results that are inferior to the average performance obtained with NAM or NAM-Quick. Although the Mixture Model outperforms usually the baseline, it does not look strong enough for novelty detection at sentence level. Overall, the NAM-Quick model is the strongest method (especially, in terms of P@10). The Mixture Model looks only competitive

	Short Queries					
	TREC 2002		**TREC 2003**		**TREC 2004**	
	P@10	**P@30**	**P@10**	**P@30**	**P@10**	**P@30**
Baseline	0.1571	0.1415	0.5660	0.4813	0.2540	0.2020
Best NAM	0.2020*	0.1680*	0.6060	0.5713*	0.2600	0.2373*
Best NAM-Quick	0.2143*	0.1721*	0.6140	0.5693*	0.2860	0.2373*
Avg. NAM	0.1914	0.1614	0.5961	0.5570*	0.2480	0.2291*
Avg. NAM-Quick	0.2025*	0.1653*	0.6003	0.5524*	0.2637	0.2282*
Mixture Model	0.1694	0.1469	0.6100	0.5560*	0.2480	0.2120

Table 2: Mixture Models approach results for short queries.

	Long Queries					
	TREC 2002		**TREC 2003**		**TREC 2004**	
	P@10	**P@30**	**P@10**	**P@30**	**P@10**	**P@30**
Baseline	0.1612	0.1483	0.5900	0.5027	0.2740	0.2200
Best NAM	0.1898	0.1667	0.6200	0.5727*	0.2760	0.2533*
Best NAM-Quick	0.2082*	0.1701*	0.6180	0.5733*	0.2980	0.2533*
Avg. NAM	0.1774	0.1581	0.5944	0.5594*	0.2686	0.2470*
Avg. NAM-Quick	0.1840	0.1613	0.6011	0.5555*	0.2816	0.2461*
Mixture Model	0.1286	0.1381	0.6220	0.5753*	0.2740	0.2373

Table 3: Mixture Models approach results for long queries.

in TREC 2003, which is a collection with high percentage of sentences that are relevant and novel. This might indicate that the LM estimation used by the Mixture Model are only effective when they are constructed from collections with high amounts of relevant material. Anyway, further analysis need to be conducted to clarify why the automatic estimation applied by this model is not effective.

5. CONCLUSIONS

In our work we address the novelty detection problem using two alternative methods. On one hand, we modify the NAM model in order to have a more efficient redundancy filtering method. We not only get a computationally efficient approach but also the performance of the model is improved.

On the other hand, we adapt and test a previous Mixture Model approach in the context of novelty detection at sentence level. We consider important that the method estimates automatically its internal parameter with the EM-algorithm. However, this model was unable to yield state-of-the-art performance. In the future, we will try to conduct further analysis to understand the reasons behind such poor performance.

6. ACKNOWLEDGEMENTS

This work was partially supported by projects TIN2005-08521-C02-01 and PGIDT07SIN005206PR.

7. REFERENCES

[1] Allan, J., Wade, C. and Bolivar, A. Retrieval and novelty detection at the sentence level. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2003)*, pages 314–321, 2003.

[2] Bernstein, Y. and Zobel, J. Redundant documents and search effectiveness. In *Proceedings of the* 14^{th} *ACM International Conference on Information and Knowledge Management (CIKM'05)*, pages 736–743, 2005.

[3] Carbonell, J.G. and Goldstein, J.. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the* 21^{th} *Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 1998)*, pages 335–336, 1998.

[4] Croft, W.B. and Lafferty, J. Language Modeling for Information Retrieval. Dordrecht: Kluwer Academic Publishers, 2003

[5] Fernández, R.T. The effect of smoothing in Language Models for novelty detection. In *Proceedings of the BCS IRSG Symposium: Future Directions in Information Access 2007*, pages 11–16, 2007.

[6] Fernández, R.T. and Losada, D.E. Novelty detection using Local Context Analysis. In *Proceedings of the* 30^{th} *Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2007)*, pages 725–726, 2007.

[7] Harman, D. Overview of the TREC 2002 Novelty Track. In *Proceedings of the* 11^{th} *Text REtrieval Conference (TREC 2002)*, 2002.

[8] Li, X. Sentence level information patterns for novelty detection. PhD. thesis. Department of Computer Science, University of Massachusetts at Amherst. 2006.

[9] Li, X. and Croft, W.B. Novelty detection based on sentence level patterns. In *Proceedings of the* 14^{st} *ACM) International Conference on Information And Knowledge Management*, pages 744–751, 2005.

[10] Ponte, J. and Croft, W.B. A language modeling approach to information retrieval. In *Proceedings of the* 21^{st} *Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 1998)*, pages 275–281, 1998.

[11] Soboroff, I. Overview of the TREC 2004 Novelty Track. In *Proceedings of the* 13^{th} *Text REtrieval*

Conference (TREC 2004), 2004.

[12] Soboroff, I and Harman, D. Overview of the TREC 2003 Novelty Track. In *Proceedings of the* 12^{th} *Text REtrieval Conference (TREC 2003)*, 2003.

[13] Xu, J. and Croft, W.B. Improving the effectiveness of Information Retrieval with local context analysis. ACM Transactions on Information Systems, vol. 18(1), pages 79–112, 2000.

[14] Xu, J. and Croft, W.B. Query Expansion using local and global document analysis. In *Proceedings of the* 19^{th} *Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 1996)*, pages 4–1, 1996.

[15] Zhai, C., Cohen, W.W. and Lafferty, J. Beyond independent relevance: methods and evaluation metrics for subtopic retrieval. In *Proceedings of the* 26^{th} *Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2003)*, pages 10–17, 2003.

[16] Zhang, Y., Callan, J. and Minka, T. Novelty and redundancy detection in adaptive filtering. In *Proceedings of the* 25^{th} *Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2002)*, pages 81–88, 2002.