Universidade de Santiago de Compostela
Departamento de Electrónica e Computación

PHD THESIS

IMPROVING SEARCH EFFECTIVENESS IN SENTENCE RETRIEVAL AND NOVELTY DETECTION

Author:
Ronald Teijeira Fernández

PhD supervisor:
David E. Losada

Santiago de Compostela, January 2011

Dr. **David Enrique Losada Carril**, Profesor Titular de Universidad en el área de Ciencia de la Computación e Inteligencia Artificial en la Universidade de Santiago de Compostela

HACE CONSTAR:

Que la memoria titulada **Improving Search Effectiveness in Sentence Retrieval and Novelty Detection** ha sido realizada por **D. Ronald Teijeira Fernández** bajo mi dirección en el Departamento de Electrónica e Computación de la Universidade de Santiago de Compostela, y constituye la Tesis que presenta para optar al grado de Doctor por la Universidade de Santiago de Compostela (programa de doctorado Interuniversitario en Tecnología de la Información del Departamento de Electrónica e Computación).

Santiago de Compostela, enero de 2011

Firmado: **Dr. David Enrique Losada Carril**
Director de la tesis

Firmado: **Dr. Francisco Fernández Rivera**
Director del Departamento de Electrónica e Computación

Firmado: **Ronald Teijeira Fernández**
Autor de la tesis

To my mother
To my brother and to my sister
To Noe

Acknowledgments

There are just a few occasions in our lives where we have the chance to acknowledge the people that really help us in succeeding and who encourage us in the good and bad situations. Here it is one of these few chances that I have to acknowledge all those people who have helped me to reach up to this point.

First of all, I want to acknowledge my supervisor, Dr. David E. Losada, the person who has made possible this work by leading it from its beginning. I have experienced that doing this thesis (and all its related work) has not been easy at all, but I have also realized that supervising can be even much more difficult. However, David has been able to make this work easier as he knew at each time the line we should follow, even if things were not as expected. I have learned many things from him: he has not only given me many advices during the development of this work but he has also taught me the correct way to do good research. Thanks to his help and supervision, I found this work enjoyable and challenging.

This PhD thesis, and all the related papers published while this work was in progress, were supported by Ministerio de Educación y Ciencia under project TIN2005-08521-C02-01, Xunta de Galicia under project PGIDT07SIN005206PR and Programa de Recursos Humanos do Plan Galego de Investigación, Desenvolvemento e Innovación Tecnolóxica de Galicia (INCITE 2006-2010) under a María Barbeito grant. Additional funds for research activities related to this thesis were supplied by Universidade de Santiago de Compostela (USC), Ministerio de Ciencia e Innovación under project TIN2010-18552-C03-03, Xunta de Galicia under projects PGIDIT06PXIC206023PN and Grupos emerxentes 2008/068 (funded by FEDER), and the Galician networks 2006/23 and 2009/61. I must also acknowledge the Council of European Professional Informatics Societies (CEPIS) and Association for Computing Machinery (ACM) for giving me grants to travel to SIGIR 2007 and SIGIR 2010 conferences.

I want to acknowledge Dr. Leif Azzopardi, who supervised my work during my two internships at Glasgow and taught me a different working methodology. Thanks to him, I felt like one more in the Department of Computing Science while I was working there. Additionally, I would also like to thank Guido - a colleague I met there and who I shared some work with - and, in general, all the members of the Information Retrieval Group at the University of Glasgow.

As a member of Precarios Galicia, an association for graduate and post gradu-

ate students, I would like to acknowledge all the work and dedication the association has done so far and still continues doing. Their members have been struggling against the current Spanish system in order to reach reasonable research conditions. Although there still exists much work to do, all the work they have been doing is really important because, without their help, many of the limitations concerning research in our Spanish system would be ignored and would hardly be solved.

I would like to thank to all my colleagues from my laboratory at the Departamento de Electrónica e Computación at the Universidade de Santiago de Compostela, with whom I enjoyed so much and celebrated some parties. I would like to name all my colleagues from my group (Grupo de Sistemas Intelixentes) but, particularly, Josito and José Carlos, the two guys who also belong to the IR@GSI team, and with whom I have deep conversations about Information Retrieval; and Cris and Fabi, who have accompanied me in my work since I started as a PhD student, with whom I shared some good experiences and some work discussions.

On the other hand, I want to acknowledge to all my friends in general (some of them colleagues at the same time), who maybe have not helped directly in this work but they were there when I needed them, and with whom I have spent many good times. Among all of them, I would like to name here Diego, Erica, Juan Ángel, Bea, Julián, Xulio, María, Enrique and Juan.

I would like also to acknowledge my girlfriend Noelia (Noe, lovingly), the woman who has filled my life of love, gladness and happiness. She has demonstrated that there are really important things in our lives, and love is one of the most important of those things. She has not only been sympathetic to me since I met her but also never doubted to help me if she was able to, and encouraged me in the good but, especially, in the bad moments. Being with her and all the things she makes me feel everyday are the best and biggest presents I can receive. She has become a really important person in my life, and I hope that our relationship will last forever and our future is full of special, good and happy moments we can share together.

All I have done so far has been the result of much work (more than one could expect at the beginning) that could not be possible at all without the help of those people who have been always close to me: my family. I would like to acknowledge my sister Pamela, my brother Elvis and, especially, my mother María Pilar, for all the things they have done for me: their patience, the help they tried to provide (even if sometimes they did not know how to support me), their encouragement, etc., and all those things everyone may need in the good but, particularly, in the bad moments. I would especially like to mention here my mother and express my gratitude for being there in every moment, for supporting my ideas even if she did not agree with me, for all her help, etc. and, overall, for being the best mother a human being can have.

I think all of them deserve at least these words:

Thank you very much!

All truths are easy to understand once they are discovered; the point is to discover them.

Galileo Galilei

Contents

List of Figures

List of Tables

Abstract

In this thesis we study thoroughly sentence retrieval and novelty detection. We analyze the strengths and weaknesses of current state of the art methods and, subsequently, new mechanisms to address sentence retrieval and novelty detection are proposed.

Retrieval and novelty detection are related tasks: usually, we initially apply a retrieval model that estimates properly the relevance of passages (e.g. sentences) and generates a ranking of passages sorted by their relevance. Next, this ranking is used as the input of a novelty detection module, which tries to filter out redundant passages in the ranking.

The estimation of relevance at sentence level is difficult. Standard methods used to estimate relevance are simply based on matching query and sentence terms. However, queries usually contain two or three terms and sentences are also short. Therefore, the matching between query and sentences is poor. In order to address this problem, we study in this thesis how to enrich this process with additional information: the context. The context refers to the information provided by the surrounding sentences or the document where the sentence is located. Such context reduces ambiguity and supplies additional information not included in the sentence itself. Additionally, it is important to estimate how important or central a sentence is within the document. These two components, the context and the centrality of the sentences, are studied in this thesis following a formal framework based on Statistical Language Models. In this respect, we demonstrate that these components yield to improvements in current sentence retrieval methods.

In this thesis we work with collections of sentences that were extracted from news. News not only explain facts but also express opinions that people have about a particular event or topic. Therefore, the proper estimation of which passages are opinionated may help to further improve the estimation of relevance for sentences. We apply a formal methodology that helps us to incorporate opinions into standard sentence retrieval methods. Additionally, we propose simple empirical alternatives to incorporate query-independent features into sentence retrieval models. We demonstrate that the incorpo-

ration of opinions to estimate relevance is an important factor that makes sentence retrieval methods more effective. In the course of our study, we also analyze query-independent features based on sentence length and named entities.

The combination of the context-based approach with the incorporation of opinion-based features is straightforward. We study how to combine these two approaches and the impact of such combination. We demonstrate that context-based models are implicitly promoting sentences with opinions and, therefore, opinion-based features do not help to further improve context-based methods.

The second part of this thesis is dedicated to novelty detection at sentence level. Because novelty is actually dependent on a retrieval ranking, we consider here two approaches: a) the perfect-relevance approach, which consists of using a ranking where all sentences are relevant (this is an ideal approach); and b) the non-perfect relevance approach, which consists of applying first a sentence retrieval method (therefore, the ranking may contain sentences that are not relevant).

We first study which baseline performs the best and, next, we propose a number of variations. One of the mechanisms proposed is based on vocabulary pruning. We demonstrate that considering terms from the top ranked sentences in the original ranking helps to guide the estimation of novelty. The application of Language Models to support novelty detection is another challenge that we face in this thesis. We apply different smoothing methods (Dirichlet and Jelinek-Mercer) in the context of alternative mechanisms to detect novelty (Aggregate and Non-Aggregate Models). Additionally, we test a mechanism based on mixture models that uses the Expectation-Maximization algorithm to obtain automatically the novelty score of a sentence.

In the last part of this work we demonstrate that most novelty methods lead to a strong re-ordering of the initial ranking. However, we show that the top ranked sentences in the initial list are usually novel and re-ordering them is often harmful. Therefore, we propose different mechanisms that determine the position threshold where novelty detection should be initiated. In this respect, we consider query-independent (a fixed position for all queries) and query-dependent approaches (cluster-based and normalized-score approaches).

Summing up, we identify important limitations of current sentence retrieval and novelty methods and, along this thesis, we propose alternative methods that are novel and effective.

Introduction

Information Retrieval (IR) deals with the representation, storage, organization of, and access to information items [BYRN99]. It can also be defined as the computer science branch that deals with finding material of an unstructured nature that satisfies an information need from within large collections (usually stored on computers) [MRS08]. Although the concept of information retrieval is very close to information seeking, the former definition indicates that it is a deeper task that includes, additionally, information structuring, organization and storage (efficiency and effectiveness are, therefore, two important components here). Well known information retrieval systems are web search engines (e.g. Google, Yahoo!, Bing, etc.) where users express their information needs as textual queries and, next, the system supplies a ranked list of links to web documents.

IR technology is present in many scenarios, such as personal computers (e.g. desktop search), enterprises (enterprise search), etc. IR systems deal usually with textual information, but other information formats such as images, audio and video can also be handled by specific retrieval applications. Because textual information seeking is the most common scenario, the literature usually refers to information retrieval and document retrieval indistinctly. However, note that they are not completely synonyms.

Document retrieval consists of retrieving documents or textual pieces of information from a document set that satisfy a given information need. The document base can be stored in a single computer (if the collection is relatively small), or distributed in multiple computers. On the other hand, an information need is usually expressed as a user query. A query is a sequence of terms that describe the user need. Usually, an information need may have different candidate queries and, moreover, a query may express different information needs (if it is not completely specified or it results ambiguous). Given a document base and a user query, a document retrieval system supplies a ranked set of documents estimated as relevant for the user need, sorted in decreasing order by their estimated relevance. This is a challenging process because, usually, users find it difficult to translate their information needs

into effective queries. Additionally, short queries are more common than long queries because users are reluctant to write more than two or three query terms (this happens especially in environments such as the web [SMHM99]). In this way, it is hard to know precisely the user need and, therefore, identifying relevant documents is not an easy task for a retrieval system.

Document retrieval systems are based on the notion of *relevance*. This concept is generally imprecise and depends on the situation or context of the retrieval task. For instance, the query *Eiffel Tower* may express different needs in different situations, such as a) when a person is planning to go to Paris and wants to know the location and admission fees of the monument; or b) when the same person is inside of Eiffel Tower and wants to know its history. The notion of relevance is therefore dependent on the location of the person who is formulating the query. Relevance could also be influenced by other contextual features such as the season, the weather, the time of the day, the user mood, etc.

Current document retrieval systems take usually two assumptions in order to simplify their retrieval algorithms: the *topical relevance assumption* and the *independent relevance assumption* [Zha02]. The former indicates that the relevance of documents may be measured considering some form of matching between the query terms and the document terms. But topicality is not the only important aspect to consider when measuring the relevance of a document. In fact, the need to go beyond topicality (i.e. considering additional information provided by context or other features) has been well-recognized in the literature [Sar70, Fro94]. The independence assumption indicates that the relevance of a document is independent of other documents. Following this, the ranking produced by a document retrieval system may contain documents at top positions that are very similar (near-duplicates) or identical (duplicates). However, **redundant information is often not desirable**. Users are often more concerned about looking for new information (novel documents) and they are less tolerant to get information they have already seen and, therefore, that they know [Har02]. This means that the retrieval system must consider the set of documents the user has already seen in order to estimate the relevance of a document. In fact, Goffman [Gof64] stressed that the relevance of a document is dependent on the previously retrieved documents. To address this problem, some sort of novelty detection mechanism must be applied. Given a ranked set of documents (e.g. the estimated relevant documents provided by a document retrieval system), novelty detection consists of filtering out documents in the ranking that provide redundant information, preserving only novel material. Formally, Li and Croft [LC08] argued that "novelty or new information means new answers to the potential questions representing a user's request or information need". This definition

involves two aspects: on one hand, a user need may be expressed by one or more questions or requirements and, on the other hand, novel information is obtained by detecting those documents that include previously unseen answers. Two alternatives are possible here depending on the kind of novel documents that users want: a) users might want to continue searching for documents related to a topic area previously found novel (directed novelty), or b) users might be interested in looking for documents that do not contain information seen before (undirected novelty) [XY08]. Directed novelty is more oriented to interactive IR systems. The retrieval is dependent on the interaction between the user and the system (the user marks a subtopic as novel and demands more material related to this subtopic). Undirected novelty is mostly focused on satisfying the original user need. In our work we only consider undirected novelty.

Novelty is useful in document retrieval. The most appropriate method would provide us the right combination of relevant documents in the top-ranked positions [WZ09]. In fact, in a real environment such as the web, users do not tend to look beyond the first top-ranked documents. Chen and Karger [CK06] stated that attempting to retrieve many relevant documents can actually reduce the chances of finding any relevant documents (because of the lack of diversity).

Some studies attempted to integrate novelty with topicality by introducing the concept of redundancy as the opposite of novelty. They defined redundancy as the amount of relevant information in a document that is covered by relevant documents delivered previously [XY08, AWB03, ZCL03, ZCM02]. Carbonell and Goldstein [CG98] attempted to combine topicality and novelty (as a non-topicality feature) to estimate the relevance of documents. Nevertheless, many authors claim that relevance and redundancy should each be modeled explicitly and separately [ZCM02].

The result of a novelty detection system is often a ranked set of documents that are both relevant and novel. Note that, because the novelty detection system is based on an input relevance ranking, the effectiveness of novelty detection is dependent, in some way, on the relevance ranking itself.

Novelty detection conforms an important module in many potential applications of other IR areas: question-answering (QA), text summarization, adaptive filtering and subtopic extraction. In QA systems, the query is a question and the answer is a reduced amount of words that respond to the query. These systems look for a brief and unique response. Therefore, novelty detection systems are useful because they process the sentences that are candidates for a given question and filter redundant material. Many text summarization systems extract the set of sentences that briefly summarize a document or a set of documents. Novelty detection is a useful module in

these systems because redundant sentences should not be considered in the summary. Adaptive filtering systems retrieve documents (or sentences) that are relevant to a user profile and do not contain redundant information with respect to previous documents (or sentences). Novelty detection estimates whether or not a document (or sentence) is novel with respect to the material already seen. Subtopic extraction systems extract all possible subtopics from a query. Given a text or a small piece of information, a novelty detection mechanism detects whether or not the text covers a subtopic tackled before, and it may also identify new generic subtopics.

Novelty detection is also related to the concept of diversity. The same query may have more than a single meaningful interpretation, and every interpretation may involve many different subtopics [ZCL03]. For instance, given the query *spirit* we may be referring to the soul, to alcoholic drinks, to the courage to do something, etc. Furthermore, each of these interpretations may involve different subtopics or *facets* [CC09]. For instance, in the example above, given the alcoholic drink interpretation, the user may be interested in the distillation process, possible brands, kinds of alcoholic drinks, etc. It is desirable that, if the system is not able to know the exact interpretation or subtopic the user is interested in, it provides answers to each of the possible interpretations/subtopics for the query. Therefore, relevant documents that cover different interpretations/subtopics are shown in top positions in the rankings so that the potential answer for the user need should be given as soon as possible. Note that, in this scenario, the utility of a document is clearly dependent on the other documents in the ranking.

The concepts of novelty and diversity are related but they are not the same. On one hand, given two documents that cover different subtopics and/or facets, it is possible that they contain information that is repeated in both of them. A diversity-oriented system will likely show both documents in the top positions in the ranking. However, a novelty detection system may consider them as redundant because they overlap. On the other hand, given two documents, they might be classified as novel but, still, they might cover the same subtopic. This difference between novelty and diversity is further discussed in [XY08].

In this work we adopt the novelty detection task as defined in the TREC 2002, 2003 and 2004 Novelty Tracks [Har02, SH03, Sob04]. The TREC Novelty Tracks divide the novelty task into two main subtasks: a sentence retrieval subtask, which consists of, given a set of queries and a set of relevant documents for each query, produce a ranked set of sentences; and a novelty detection subtask, which consists of filtering out redundant sentences from this ranking. Considering novelty detection at document level may be problematic because nearly every document contains something new, particularly

when the domain is news[1] [SH05]. To address this problem, the novelty task was defined at sentence level. Sentences are short pieces of information with a semantic and lexical structure that, unlike documents, are characterized for providing a short idea or concept in a concise way. Therefore, considering sentences as pieces of information is a natural way to study novelty.

The sentence retrieval task consists of finding relevant sentences from a document base given a query. This task is very useful in a wide range of Information Retrieval applications, such as summarization, novelty detection, question answering and opinion mining. Sentence retrieval is a challenging problem area that has attracted a great deal of attention recently [AWB03, WJR05, Mur06, LF07, Los08]. The bulk of sentence retrieval methods proposed in the literature are a straightforward adaptation of standard retrieval models (such tf-idf, BM25, Language Models, etc), where the sentence is the unit of retrieval, as opposed to the document. This leads to sentence retrieval models which estimate relevance based only on the match between query and sentence terms.

In this thesis we study and propose different methods to support the effective retrieval of relevant and novel sentences. On one hand, we define, implement and evaluate different approaches to address the sentence retrieval problem. This includes a thorough comparison between state of the art measures. First, we introduce query-independent features that help to estimate the relevance of sentences. In this study we consider features based on the presence of named entities, the presence of opinions and the length of sentences. Remarkably, we analyze and employ successfully opinion-based information in sentence retrieval, which is a novel contribution in this area. Our proposed query-independent features help to improve current state of the art sentence retrieval methods with no significant computational penalties. Next, we consider that sentences are not isolated pieces of information, i.e. they are usually dependent on a context. This context usually comes from the closest sentences or from the document as a whole. Therefore, we propose a formal approach, based on Statistical Language Models, to model this context in a standard sentence retrieval setting.

The second part of this thesis is dedicated to study novelty detection given a ranking of sentences. To this aim, we make a deep analysis of current standard novelty detection methods and design new effective mechanisms given two different scenarios: a perfect relevance scenario, where we start from a ranked set of sentences judged as relevant by the assessors; and a non-perfect relevance scenario, where we employ a ranking of estimated relevant sentences

[1]TREC Novelty datasets contain documents that are news extracted from different information sources.

provided by a standard sentence retrieval mechanism. First, we analyze the performance of current state of the art novelty detection mechanisms and propose variants of these methods that consist of applying vocabulary pruning in order to focus the novelty process solely on on-topic sentences. We also propose novel length-based normalizations for current novelty detection methods. Next, we propose more formal approaches, based on Statistical Language Models, that model sentences as probability distributions and estimate novelty with the divergence between such distributions. In the course of this study, we also analyze a two-mixture model that estimates novelty with automatic parameter estimation (Expectation-Maximization algorithm). Finally, we demonstrate that the perfect relevance scenario is harder to improve than the non-perfect relevance scenario. Therefore, we focus on the perfect relevance case and propose new novelty detection mechanisms based on freezing the top-ranked sentences.

Contributions

In this thesis we conduct a complete analysis of the TREC Novelty Track and analyze thoroughly the problems presented in this scenario.

Regarding sentence retrieval, our main contributions are:

- A comparative study of performance of different standard sentence retrieval models, such as tfisf, BM25 and methods based on Language Models.
- A proposal of novel query-independent features for sentence retrieval: opinion-based features, features based on named entities and sentence length.
- The successful application of a formal methodology to include query-independent features into existing sentence retrieval models: Feature's Logs-Odd Estimator (FLOE). This includes an study of the combination of different query-independent features. By incorporating these features into standard retrieval models we obtain significant improvements in performance. In particular, the effect of opinion-based features on performance is highly beneficial.
- A thorough study of sentence retrieval in the framework of Language Models.
- The incorporation of local context (document and surrounding sentences) into methods based on Language Models. This leads to novel

and formal approaches that are able to outperform state of the art methods.

- The incorporation of sentence importance into sentence retrieval models following a Language Modeling approach. The inclusion of sentence importance into retrieval models leads to substantial gains.

- A study of the combination of context and opinion-based information in order to estimate the relevance of sentences.

Regarding novelty detection, our main contributions are:

- A study of novelty detection in different scenarios: perfect relevance and non-perfect relevance.

- The evaluation of current state of the art novelty detection methods against competitive baselines.

- A study of the impact of vocabulary pruning on standard novelty methods. In order to get the vocabulary, two different mechanisms are considered: Local Context Analysis and Divergence From Randomness. We show the conditions that make this variant outperforms the original models.

- The evaluation of formal methods in the context of Language Models to address novelty: Aggregate (AM) and Non-Aggregate (NAM) models, which use Kullback-Leibler Divergence (KLD). AM considers the set of previously seen sentences as a whole and NAM makes pair-to-pair comparisons between a sentence and each of the previously seen sentences. A complete comparative study of this kind had not been conducted in the literature.

- The proposal of an effective and efficient variant of the NAM model: NAM-Quick. This model is similar to NAM but, instead of using KLD, it employs a modified version of KLD. This variant performs at least as well as the original version and it is much more efficient.

- The application of a mixture model that combines a background model, a reference model and a model for the sentence in order to detect novelty. We use the Expectation-Maximization algorithm to estimate automatically the parameters.

In the course of our study, all the avenues followed revealed that novelty detection is an extremely challenging task where it is highly difficult to beat naive baselines. We therefore propose further variants of standard methods to improve effectiveness:

- Methods based on freezing the top ranked sentences and re-ordering the remaining sentences with a standard novelty detection mechanism. To this aim, a query-independent threshold (fixing the same threshold for all queries) and a query-dependent thresholds are considered. For query-dependent thresholding, cluster-based and score-based approaches are considered. We demonstrate that it is better to freeze the first positions and start detecting novelty at lower positions. This is a novel contribution to the information retrieval community in this area.

Publications Derived from this Thesis

The list of publications derived from this thesis is:

- Ronald T. Fernández, David E. Losada. *Using Opinion-Based Features to Boost Sentence Retrieval*. Published as a short paper in the proceedings of the ACM 18^{th} Conference on Information and Knowledge Management (CIKM 2009) (short papers acceptance rate: 20.2%) [FL09]: In this work, we provided experimental evidence to show that the subjectivity of a sentence, the number of terms with negative orientation and the number of opinionated terms are sentence features that help to estimate relevance. The use of opinion-based features in sentence retrieval was a novel contribution and, additionally, we opened up a new line of research: leveraging different forms of prior information in order to improve baseline retrieval.

- Ronald T. Fernández, David E. Losada, Leif A. Azzopardi. *Extending the Language Modeling Framework for Sentence Retrieval to Include Local Context Information Retrieval*. Published in Information Retrieval Journal (JCR journal with 1.8 of impact factor in 2009) [FLA10]: In this work, we proposed several novel probabilistic Language Models to address the sentence retrieval problem by including the local context: a) localized smoothing, in order to provide a better estimate of the probability of a term in a sentence, and b) importance of sentences within the document, i.e. the centrality of a sentence in the document. With both forms of local context, we significantly outperformed the standard Language Modeling approach applied to sentence retrieval and the current state of the art sentence retrieval models.

- Leif Azzopardi, Ronald T. Fernández, David E. Losada. *Improving Sentence Retrieval with an Importance Prior*. Published as a poster in proceedings of the 33^{rd} ACM International Conference on Research and Development in Information Retrieval (SIGIR 2010) (poster acceptance rate: 30.7%) [AFL10]: We proposed and empirically evaluated an extension of the Language Modeling framework for sentence retrieval to include sentence importance through a prior. By including this prior, substantial improvements were obtained for all the different Language Models, which resulted in significantly better performance. This work also suggests that the naive application of document retrieval models to other task may lead to non-optimal performance.

- Ronald T. Fernández, David E. Losada. *Novelty Detection Using Local Context Analysis*. Published as a poster in proceedings of the 30^{th} ACM International Conference on Research and Development in Information Retrieval (SIGIR 2007) (poster acceptance rate: 44.7%) [FL07]: In this work we presented the results of our attempts to identify relevant and novel sentences in a ranked list of documents using different methods and their variants using the Local Context Analysis (LCA). Given the state of the art novelty detection methods, our results indicated that precision at top ranks might be further improved if redundancy decisions are made in terms of a more focused vocabulary.

- Ronald T. Fernández. *The Effect of Smoothing in Language Models for Novelty Detection*. Published as a full paper in proceedings of Future Directions in Information Access (FDIA 2007) [Fer07]: We studied novelty detection at sentence level using smoothed Language Models and Kullback-Leibler Divergence. We focused our work on applying different smoothing techniques (Dirichlet and Jelinek-Mercer) with varying parameter setting and, in order to study novelty, we applied two different techniques to build the Language Models (Aggregate and Non-Aggregate Models). We thoroughly compared the performance of the different methods and settings.

- Ronald T. Fernández, David E. Losada. *Novelty as a Form of Contextual Re-ranking: Efficient KLD Models and Mixture Models*. Published as a full paper in proceedings of the 2^{nd} Information Interaction in Context (IIiX 2008) [FL08]: We addressed novelty detection problem with two alternative methods based on Language Models: a) we modify the Non-Aggregate Model in order to have a more efficient redundancy filtering method, and b) we adapted and tested a previous mixture model approach in the context of novelty detection at sentence level, which

estimates automatically its internal parameters with the Expectation-Maximization algorithm.

- Ronald T. Fernández, Javier Parapar, David E. Losada, Álvaro Barreiro. *Where to Start Filtering Redundancy? A Cluster-Based Approach*. Published as a poster in proceedings of the 33^{rd} ACM International Conference on Research and Development in Information Retrieval (SIGIR 2010) (poster acceptance rate: 30.7%) [FPLB10]: We focused this work on novelty detection given a perfect relevance environment. We analyzed the performance of current state of the art novelty detection methods and demonstrated that, usually, these methods work worse than doing nothing. Therefore, we proposed a mechanism that consists of starting the novelty detection process from a given rank position. To this aim, we followed a query-dependent cluster-based method that predicts a good novelty detection starting position. We showed that statistically significant improvements between this variant and the state of the art novelty detection methods were obtained.

- Rafael Berlanga, Aurora Pons, David E. Losada, Ronald T. Fernández. *Recuperación de Información: un enfoque práctico y multidisciplinar*, chapter *Técnicas Avanzadas de Recuperación de Información II* [BPLF11]: This is a chapter in a teaching book (which is being edited) about topic detection and tracking, novelty detection and automatic summarization.

Chapter 1

Sentence Retrieval

The sentence retrieval (SR) task consists of finding relevant sentences from a document base given a query. This task is very useful in a wide range of Information Retrieval (IR) applications, such as summarization, novelty detection, question answering and opinion mining. SR is a challenging problem area that has attracted a great deal of attention recently [AWB03, WJR05, Mur06, LF07, Los08]. The bulk of SR methods proposed in the literature are a straightforward adaptation of standard retrieval models (such tf-idf, BM25, Language Models, etc), where the sentence is the unit of retrieval, as opposed to the document. This leads to SR models which estimate relevance based only on the match between query and sentence terms. In fact, the state of the art SR method is known as term frequency-inverse sentence frequency (tfisf) which is analogous to the traditional tf-idf method used in document retrieval [AWB03, Los08]. While, numerous attempts to develop more sophisticated models that employ techniques, such as Natural Language Processing and Clustering have been proposed [LC05, KSC+03, ZLL+03], they have failed to significantly and consistently outperform the tfisf method. Consequently, little progress has been made in terms of improving sentence retrieval effectiveness.

The best performing sentence retrieval techniques attempt to perform matching directly between the sentences and the query. However, in this thesis we go a step further and propose two different mechanisms to enhance the performance of current state of the art sentence retrieval methods:

- **Incorporating query-independent features into standard sentence retrieval models:** To meet this aim, we apply a formal methodology and consider query-independent features of different nature. In particular, we show that opinion-based features are promising. Opinion mining is an increasingly important research topic but little is known

about how to combine opinion-based information with standard retrieval scores. We consider here different kinds of opinion-based features to act as query-independent evidence. On the other hand, the length of the retrieval unit has been demonstrated to be an important component in different retrieval scenarios. We therefore include other features in our study, such as sentence length or named entities. Our evaluation demonstrates that, either in isolation or in combination, these query-independent features help to improve substantially the performance of state of the art sentence retrieval methods.

- **Incorporating the local context of a sentence into existing sentence retrieval models:** Using a Language Modeling framework, we propose a novel reformulation of the sentence retrieval problem that extends previous approaches so that the local context is seamlessly incorporated within the retrieval models. In a series of comprehensive experiments, we show that localized smoothing and the prior importance of a sentence can improve retrieval effectiveness. The proposed models significantly and substantially outperform the state of the art and other competitive sentence retrieval baselines on recall-oriented measures, while remaining competitive on precision-oriented measures. We demonstrate that local context plays an important role in estimating the relevance of a sentence, and that existing sentence retrieval Language Models can be extended to utilize this evidence effectively.

The two research lines sketched above are complimentary mechanisms to deal with the poor matching between the query and sentences that usually occurs in sentence retrieval. In the final part of this chapter, we study the combined use of these strategies.

1.1 Related Work

In the sentence retrieval literature, most of the proposals consist of addressing the SR problem by adapting document retrieval methods with little change. We believe that this is not the most appropriate approach because the peculiarities of the task are largely ignored. Sentences are short pieces of information. Most sentence retrieval methods are based on a regular matching between query and sentences. However, sentences that do not contain query terms may be relevant for a query. Query expansion is a mechanism that tries to address this problem. This is a way to avoid the vocabulary mismatch problem, which is rather severe in sentence retrieval. The study reported in [Los10] analyzes carefully different query expansion methods applied to

sentence retrieval. This included well-known term selection techniques, such as those based on regular pseudo-relevance feedback and Local Context Analysis [XC96, XC00], and two different expansion configurations: before and after sentence retrieval. The paper concludes that the ideal expansion configuration depends strongly on the quality of the initial query. Evolved expansion methods, based on selective feedback, were studied in [JAC+04]. They are more stable than standard feedback methods but require training data. On the other hand, other authors resort to lexical expansion, i.e. they utilize query-related terms (i.e. synonyms or related terms from a lexical resource) to expand the query. This approach may not be appropriate because noisy terms are likely introduced into the expanded query [Voo93] and, moreover, a large terminological resource is not always available.

Given the inconsistent effects on performance and the time requirements involved at query time, query expansion is problematic for sentence retrieval. We therefore take here a different avenue to address retrieval problems at sentence level. We reckon that the estimation of relevance could be more accurate by using query-independent information. In the literature, there is not much evidence about the combination of query-dependent and query-independent information to estimate relevance for SR problems. We consider some opinion-based features and study whether or not they help to improve sentence retrieval performance. We also include other features, such as name entities and sentence length, in our study. Additionally, we analyze whether the combination of features of the same or different nature improves performance over individual incorporations.

The use of opinions for sentence retrieval was also applied in [KRH04]. For the TREC 2003 and 2004 opinion topics, relevant opinion sentences were recognized using opinion-bearing word lists. However, the authors assumed that opinion-based methods are only effective for opinion topics. We demonstrate here otherwise. Furthermore, the performance achieved by the methods described in [KRH04] was not higher than the performance of state of the art methods. Unfortunately, the experiments reported in [KRH04] cannot be replicated here because they are based on collecting manually opinion-bearing words from resources such as WordNet. Since the manual lists are not publicly available we cannot incorporate this approach into our experimental study. In [LC08], Li and Croft considered different opinion-based features to estimate the novelty nature of sentences. As a first stage, they re-rank sentences considering opinion-based patterns and, next, they filtered out redundant sentences by applying a novelty detection method. This is a way to study the impact of opinion-based features on novelty detection. In contrast, we focus here on sentence retrieval and apply a methodology that is totally different to Li and Croft's one.

One of the first sentence retrieval methods in the literature was coined as tfisf [AWB03]. It is an adaptation of tf-idf in document retrieval for sentence retrieval. This simple approach is regarded as the state of the art in SR as it has proved to consistently outperform other methods [AWB03, LF07, FL09]. As a matter of fact, this parameter-free method has been shown to perform at least as well as the best performing empirically tuned and trained SR models based on BM25 or Language Models (LMs) [LF07, FL09]. While this tends not to be the case in document retrieval, on other tasks where the unit of retrieval is smaller such as passage retrieval, vector-space models have performed empirically well. For instance, Kaszkiel and Zobel [KZ97, KZ01] showed that some cosine and pivoted models are highly effective for document ranking based on passages. Although we evaluate here SR (rather than document retrieval), past studies on passage-based document retrieval confirm also that vector-space methods are also state of the art models for query-passage scoring. This further supports our choice to select a vector-space measure as our baseline.

Clustering methods have been also considered as alternative techniques to improve SR models, but such methods have shown mixed performance [KSC+03, ZLL+03] seldom improving upon the tfisf baseline. Besides, these clustering methods also incur additional computation costs and increased complexity making them unattractive to implement.

Besides applying query-independent features, we reformulate the problem of sentence retrieval within the Language Modeling framework, where localized smoothing is employed to improve the representation of sentences. The work most related to this research has been performed by Losada and Fernández [LF07] and Murdock [Mur06]. In [LF07], the local context of a sentence was informally introduced into the computation of sentence similarity. Basically, extra weight was given to those terms that have high frequency in the associated documents. In [Mur06], the estimation of the sentence Language Model included some local context, and combines the evidence from the sentence and document level. More specifically, a simple mixture model of the sentence, document and collection was proposed in order to form a better representation of the sentence. From the limited experiments reported, Murdock showed that the mixture model was better than other LM methods for the TREC novelty data. However, the results are far from conclusive because competitive SR methods, such as tfisf, were not evaluated. Nor was any indication of the sensitivity of the method with respect to the smoothing parameters reported. We provide here a more general framework that encompasses both previous formulations using Language Models, but also provides avenues for incorporating other forms of local context.

In the following subsections we introduce, first, the collections and exper-

Task 1: Given the set of documents for the topic, identify all relevant and novel sentences.
Task 2: Given the relevant sentences in all documents, identify all novel sentences.
Task 3: Given the relevant and novel sentences in the first 5 documents only, find the relevant and novel sentences in the remaining documents.
Task 4: Given the relevant sentences from all documents and the novel sentences from the first 5 documents, find the novel sentences in the remaining documents.

Figure 1.1: Tasks defined in the TREC 2003 and TREC 2004 Novelty Tracks.

imental settings utilized in our experiments. Second, we use opinion-based features, name entities and sentence length as query-independent features in order to improve the performance of current retrieval methods. Third, we employ the information provided by the document or the surrounding sentences as a context for sentences. Finally, we combine both mechanisms, i.e. query-independent features and document or surrounding sentences information and analyze the resulting models.

1.2 TREC Novelty Track

Novelty Detection at sentence level was formally defined in the TREC Novelty Track, in the TREC 2002 [Har02], 2003 [SH03] and 2004 [Sob04] evaluation conferences. This task is divided into two subtasks: sentence retrieval and novelty detection. In this thesis, we adopt the novelty task as defined in the TREC Novelty Tracks and study carefully both subtasks trying to design effective methods that support them.

The groups participating in this track start from a common ranking of documents for each query. The sentence retrieval task consists of obtaining a ranked set of estimated relevant sentences given the ranked documents. The novelty detection task aims at filtering out redundant sentences from the ranked set of sentences. Therefore, the aim of this task is to obtain a ranked list of sentences where the top positions are occupied by relevant sentences that are also novel. For those familiar with the Novelty Track, in this thesis we are concerned with Task 1 and Task 2 of the track (see Figure 1.1).

The notion of novelty or new information was defined in this context as new answers to the potential questions representing a user's request or information need [Li06, LC08]. The assumption is that, initially, the user does

not know any potential answers to his/her need and that all the knowledge he/she acquires comes from the material that the retrieval system supplies. Given this assumption, novel sentences are those ones that satisfy the user need and do not include previously seen information. The user's information need is represented with a textual query, consisting on a few keywords.

In our evaluation we considered the TREC novelty datasets in 2002 [Har02], 2003 [SH03] and 2004 [Sob04]. These test collections supply relevance and novelty judgments at sentence level for each topic. The features for each collection are described as follows:

TREC 2002 : Among an initial set of 150 topics extracted from earlier TRECs (TREC 6, 7 and 8) - topics 300-450 - assessors selected those ones that had between 10 and 70 relevant documents and eliminated a few that had large numbers of Federal Register documents (which tend to be very long). As a result, 50 topics[1] were selected and a maximum of 25 relevant documents for each topic.

TREC 2003 : In this collection topics were built specifically by assessors for this task. This resulted in a set composed of 50 topics, where 28 of them are event topics and the remaining 22 are concerned opinions about controversial subjects. Next, given the AQUAINT dataset, a retrieval system named WebPRISE was used in order to get a set of documents that answer to each topic. For each query, assessors collected 25 relevant documents from this set and sorted them in chronological order[2].

TREC 2004 : Just like in TREC 2003, topics were build specifically by assessors for this track. This year, 25 of the topics were events and the remaining 25 were opinion topics. Documents were obtained following the same procedure as in the previous year. The only difference with respect to the previous year's dataset is that some irrelevant documents that were close matches to relevant ones were also included [Sob04].

In order to find the set of relevant sentences for each topic, sentence-tagged documents were supplied to participants (see Figure 1.2). The judgment process was as follows. Given a topic expressed as a query, all sentences

[1] One assessor disagreed with the original assessor's relevance judgments for a topic and could not find relevant sentences in any of the documents. Therefore, this topic was removed in the relevance and novelty judgments.

[2] The reason for this sort is that, as documents contain news, the background information tends to occur more completely in earlier articles and it is summarized more briefly as time goes on and new information is reported.

```
<DOC>
<DOCNO>

<s docid="XIE20000821.0014" num="1"> XIE20000821.0014</s>
</DOCNO>
<DATE_TIME>

<s docid="XIE20000821.0014" num="2"> 2000-08-21</s>
</DATE_TIME>
<BODY>
<HEADLINE>

<s docid="XIE20000821.0014" num="3"> French Defense Minister Considers Sinking of
Kursk Accident of Maneuver</s>
</HEADLINE>
<TEXT>
<P>

<s docid="XIE20000821.0014" num="4"> PARIS, August 21 (Xinhua) -- French Defense
Minister Alain Richard said on Monday that the sinking of the Russian nuclear-
powered submarine Kursk was "an accident of  exercise and maneuver" rather than
the result of a collision with another submarine.</s>
</P>
<P>

<s docid="XIE20000821.0014" num="5"> In an interview with radio Europe 1, Richard
said that at the time of the exercise of the Russian Northern Fleet in the Barents
Sea, there were only Russian warships
present.</s>
</P>
<P>

<s docid="XIE20000821.0014" num="6"> "My conviction is that this is very probably
an accident of exercise, of maneuver, in a surrounding where there were only
Russian military ships," he said.</s>
</P>
<P>

<s docid="XIE20000821.0014" num="7"> Richard also pointed out that there were
conventional real shootings during the exercise.</s>
</P>
<P>

<s docid="XIE20000821.0014" num="8"> Russian officials have repeatedly suggested
that "Kursk" could have collided with a British submarine before it sank.</s>
</P>
</TEXT>
</BODY>
</DOC>
```

Figure 1.2: Example of a sentence-tagged document (extracted from the TREC 2003 Novelty dataset).

were evaluated sequentially and marked as relevant by the assessors (if they provided information requested by the query) or non-relevant (otherwise). The list of relevant sentences was built by taking the set of relevant sen-

tences preserving their original order, i.e. documents preserve the natural order provided by NIST and multiple sentences from the same document are considered in the order in which they appear in the document. The average percentage of relevant sentences in TREC 2003 and TREC 2004 is 41.1% and 19.2%, respectively. In TREC 2002 only 2% of the sentences were estimated as relevant. Given this marginal amount of relevant sentences, this collection is not appropriate for estimating redundancy because nearly all relevant sentences are novel. The statistics of novel material constructed upon this data would not be reliable. As a matter of fact, the characteristics of the TREC 2002 Novelty Track data have been criticized in the past and TREC 2003 and TREC 2004 are regarded as more robust novelty benchmarks [Li06]. We will therefore use TREC 2002 as a hard sentence retrieval benchmark (2% relevant sentences) but we will not use it for novelty detection because of the lack of redundancy. Given the set of relevant sentences, the overall percentage of novel sentences in TREC 2002, 2003 and 2004 is 90.9%, 65.7% and 41.4%, respectively.

In our experiments we evaluated only short queries (built considering only the TREC title field - see Figure 1.3) because they are by far the most utilized ones (especially in environments such as the web) [SMHM99]. Observe that we use short queries while the teams participating in the TREC Novelty Tracks were allowed to use the whole topic. This means that the results presented here are not directly comparable to the official TREC results. In our preprocessing we removed stopwords[3] and did not apply stemming[4]. The statistics for these datasets are shown in Table 1.1.

	#topics	**#event topics**	**#opinion topics**	**#novel sents.**	**#relevant sents.**	**#total sents.**
TREC 2002	49	N/A	N/A	1241 (90.92% of rels)	1365 (2.4%)	57227
TREC 2003	50	28	22	10226 (65.73% of rels)	15557 (39.07%)	39820
TREC 2004	50	25	25	3454 (41.40% of rels)	8343 (15.97%)	52257

Table 1.1: Statistics for TREC 2002, 2003 and 2004 datasets.

The F measure was the official measure of performance in the TREC Novelty Track. However, this measure is not very precise in characterizing the

[3]The list of 571 stopwords used in our experiments is available in Appendix A.

[4]We made preliminary experiments with different preprocessing configurations and found no major differences. Still, removing stopwords and applying no stemming was slightly superior to other preprocessing alternatives.

```
<top>
<num> Number: N33
<title> Russian submarine Kursk sinks
<toptype> event

<desc> Description:
The Russian submarine Kursk sank in the Barents Sea killing
all 118 aboard in August 2000.

<narr> Narrative:
Reports on what was known about the sinking of the Russian nuclear
powered submarine, Kursk, are relevant.  Speculation about
what caused the explosions aboard; description of the vessel and its
capabilities, and mention of efforts to rescue the crew are relevant.
Reports that U.S. submarines were monitoring Russian navy exercises
and Russia's suspicions that the Soviet submarine K-128 was struck
by an American submarine and sunk in 1968 are relevant. Mention of
the fact that Russia turned down a U.S. offer to send a deep-diving
rescue vessel is relevant.  Discussion of U.S. plans to retire one
of its two rescue vessels is not relevant.  Polls reporting how
Russians felt about the disaster and mention of ceremonies for
the dead are relevant.
</top>
```

Figure 1.3: Example of a TREC topic and its fields (extracted from the TREC 2003 Novelty dataset).

real requirements of users [LC08]. Therefore, we considered other measures to evaluate performance: P@10 and MAP.

In the sentence retrieval stage, the interpretation of these measures is straightforward. P@10 is the proportion of retrieved sentences that are relevant in the top 10 ranked sentences, i.e.:

$$\text{P@10} = \frac{\#(\text{relevant sentences retrieved in the top 10})}{10} \tag{1.1}$$

Given a set of queries, their respective P@10 values are averaged out to get a single P@10 figure.

MAP (Mean Average Precision) provides a single-figure measure of quality across recall levels. For a single information need, average precision is the average of the precision value obtained for the set of top k sentences existing after each relevant sentence is retrieved. This value is then averaged over queries [MRS08], i.e.:

$$\text{MAP} = \frac{1}{|Q|}\sum_{j=1}^{|Q|}\frac{1}{m_j}\sum_{k=1}^{m_j}\text{Precision}(R_{jk}) \tag{1.2}$$

where, given the set of relevant sentences for a query $q_i \in Q$, R_{jk} is the set of ranked retrieval results from the top result until you get to sentence s_k, m_j is the number of relevant documents for query q_j, and

$$\text{Precision}(R_{jk}) = \begin{cases} \frac{\#(\text{relevant sents. retrieved in } R_{jk})}{|R_{jk}|} & \text{, when } s_k \text{ is relevant} \\ 0 & \text{, otherwise} \end{cases} \tag{1.3}$$

In the novelty detection stage, given an ordered set of sentences, P@10 measures the percentage of sentences that are novel in the top 10. On the other hand, MAP (Mean Average Precision) is the mean of the precision scores obtained after each novel sentence is retrieved[5].

In all our experiments, statistical significance was estimated using the paired t-test[6] at confidence levels of 95% and 99% (marked with $*$ and †, respectively).

1.2.1 Problems of the Novelty Datasets

There are some concerns about different aspects of the experimental setting established by the TREC Novelty Track [BZ05]:

- Assessments: The novelty of a sentence is dependent on such sentence and the previously seen sentences. This means that the order is an important factor at the time of estimating the novelty score of a sentence. Furthermore, it is needed that the evaluation is done on a small and predefined set of documents.

- Relevant documents: In TREC 2002 and TREC 2003, all the documents considered for each query are relevant. Although this helps to study novelty with no interferences coming from non-relevant material (study novelty separately from relevance), this may introduce a bias in the final results. These results may not, therefore, predict performance in more realistic search environments [AWB03].

[5]Observe that the notion of relevance in the novelty detection stage is associated to a relevant and novel sentence.

[6]t-test was shown in the literature to be the a robust non-parametric significance test for information retrieval [SAC07].

- Overlooked information: Novelty and relevance are difficult to detect with a computer. Low precision and recall values is usually obtained by systems designed for such purpose. In the case of novelty detection, systems may identify sentences which are novel (and relevant at the same time) as redundant. Such novel information might be really important for the user but, because it has been classified as non-novel, the system is overlooking it. This inaccuracy is somewhat acceptable for relevance but it is less so for novelty (critical information might be overlooked).

- Assessors disagreement: Human judgments disagree in some sentences, i.e. sentences that were judged as novel by an assessor were judged as non-novel by another assessor, and vice versa. The difficulty of detecting novelty by humans demonstrates how difficult the task is and, thus, how hard the design of accurate novelty detection systems is.

- Authority: The number of documents (sentences) that answer to a query may help to reinforce the information provided by the system. For instance, if a user wants to know an opinion about a news article, if all the opinions come from the same source then it may not be reliable. Nevertheless, if the system provides the same or different opinions from different sources, the opinions are reinforced by each other. Therefore, although novelty detection is usually desirable, the use of these systems are dependent on the scope of the application and its use may not always be advantageous.

Despite these problems, the TREC Novelty Track has been recognized as a standard benchmark for evaluating novelty detection methods at sentence level and the collections have been re-used frequently to evaluate different sentence retrieval and novelty algorithms.

1.3 Standard Sentence Retrieval Methods

Different sentence retrieval methods were proposed in the literature. Among them, tfisf [AWB03] has been considered as the state of the art sentence retrieval method in the past [AWB03, FLA10, LF07]. Consequently, in this thesis we consider tfisf as our sentence retrieval baseline. tfisf is is an adaptation of tf-idf at sentence level:

$$sim_{\text{tfisf}}(s,q) = \sum_{t \in q \cap s} \log(c(t,q)+1) \cdot \log(c(t,s)+1) \cdot \log\left(\frac{N+1}{0.5+sf(t)}\right) \quad (1.4)$$

where s and q are a sentence and a query, respectively, $c(t,q)$ and $c(t,s)$ are the number of occurrences of term t in q and s (respectively), N is the number of sentences in the collection, and $sf(t)$ is the number of sentences that contain t.

In this section, we provide empirical evidence to demonstrate that this is a very competitive baseline. To this aim, we compare here tfisf against other popular sentence retrieval methods, such as Okapi BM25 [RWJ+94] and a Language Modeling (LM) approach (with Dirichlet smoothing) based on the Kullback-Leibler divergence (KLD), as described in [LAC+02]. BM25 can be straightforwardly applied to the SR case, such that:

$$sim_{\text{BM25}}(s,q) =$$

$$\sum_{t \in q \cap s} \log \frac{N - sf(t) + 0.5}{sf(t) + 0.5} \cdot \frac{(k_1+1)c(t,s)}{k_1\Big((1-b) + b\frac{c(s)}{avsl}\Big) + c(t,s)} \cdot \frac{(k_3+1)\cdot c(t,q)}{k_3 + c(t,q)} \quad (1.5)$$

where N is the number of sentences in the collection, $sf(t)$ is the number of sentences that contain t, $avsl$ is the average sentence length, $c(t,s)$ and $c(t,q)$ are the number of occurrences of t in the sentence s and the query q (respectively), $c(s)$ is the number of terms in s, and k_1, b and k_3 are parameters.

BM25 depends on three parameters: k_1, which controls term frequency; b, which is a length normalization factor; and k_3, which is related to query term frequency. We fixed k_3 to 0 (observe that we work with short queries and, therefore, the effect of k_3 is negligible) and experimented with k_1 ranging from 1.0 to 2.0 and b ranging from 0 to 1 (both of them in steps of 0.1).

On the other hand, the LM approach using KLD with Dirichlet smoothing is defined as:

$$sim_{\text{KLD}}(s,q) = \sum_{t} p(t|\theta_s) \cdot \log \frac{p(t|\theta_s)}{p(t|\theta_q)} \quad (1.6)$$

where θ_s and θ_q are Language Models for s and q. These models can be smoothed by applying a smoothing mechanism, such as Dirichlet smoothing [ZL04]:

$$p(t|\theta_x) = \frac{c(t,x) + \mu \cdot p(t|\mathcal{C})}{c(x) + \mu} \quad (1.7)$$

	tfisf	**BM25**	**KLD**
		TREC 2002	
P@10	.2041	.2082	.1633*
$\Delta\%$		$(+2.01)$	(-19.99)
		$(k_1 = 1.2, b = 0.0)$	$(\mu = 2500)$
MAP	.1094	.1102	.0934*†
$\Delta\%$		$(+0.73)$	(-14.63)
		$(k_1 = 1.4, b = 0.0)$	$(\mu = 500)$
		TREC 2003	
P@10	.7480	.7540	.7160*
$\Delta\%$		$(+0.80)$	(-4.28)
		$(k_1 = 1.1, b = 0.0)$	$(\mu = 250)$
MAP	.3851	.3852	.3640*†
$\Delta\%$		$(+0.03)$	(-5.48)
		$(k_1 = 1.4, b = 0.0)$	$(\mu = 500)$
		TREC 2004	
P@10	.4300	.4380	.4160
$\Delta\%$		$(+1.86)$	(-3.26)
		$(k_1 = 1.0, b = 0.0)$	$(\mu = 100)$
MAP	.2358	.2370*	.2236*†
$\Delta\%$		$(+0.51)$	(-5.17)
		$(k_1 = 1.0, b = 0.0)$	$(\mu = 250)$

Table 1.2: tfisf vs. optimal BM25/KLD models.

where $\mathcal{C}$ is the collection and μ is the parameter of smoothing. We experimented here with the following values of μ: 1, 5, 10, 25, 50, 100, 250, 500, 1000, 2500, 5000 and 10000. Observe that we optimize BM25 and KLD parameters while this luxury is not afforded to tfisf (because it is parameter-free).

Results are reported in Table 1.2 and Figure 1.4. Statistically significant differences between the baseline and BM25 or LMs with KLD at confidence levels of 95% and 99% are indicated with $*$ and †, respectively. For BM25 and KLD, the Table reports the best performance achieved in every collection and the optimal parameter settings are reported in brackets. Only in TREC 2004 (MAP) BM25 is statistically significant better than tfisf. Anyway, this corresponds with a percentile improvement which is less than 1% in performance (from .2358 to .2370) and, therefore, it is unlikely to be noticeable to the final users. In the remaining cases, tfisf performs at least as well as tuned BM25. Moreover, note that tfisf and BM25 perform better than KLD,

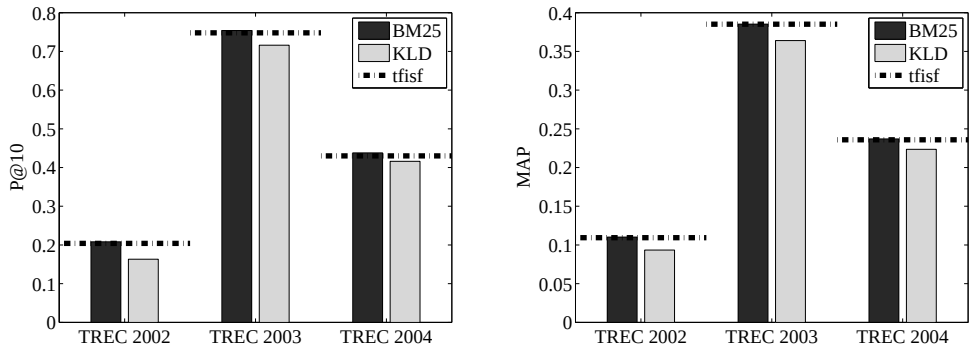


Figure 1.4: tfisf vs. optimal BM25/KLD models.

obtaining statistically significant improvements in most of cases. These results demonstrate that tfisf is an effective method that performs similarly to an optimal BM25 model. Unlike BM25, tfisf is parameter-free, which is an added value. Similar results were achieved in [Los08, LF07], where tfisf was compared against similar SR models, obtaining equivalent results.

1.4 Query-Independent Evidence for Sentence Retrieval

Many of the approaches proposed in the SR literature are direct adaptations of document retrieval methods. These methods are usually based on matching query and sentence terms. Nevertheless, sentences are very short pieces of text and, therefore, there are usually very few matching terms. Some researchers tried to alleviate this problem by applying query expansion. However, we take here an alternative approach focused on combining query-independent evidence (related to the sentences) with sentence retrieval scores, leading to effective estimations of the relevance of sentences. More specifically, we consider opinion-related information, name entities and the sentence length as query-independent features. Our intuition is that, in many situations, users are mostly interested in subjective material. This is usually the case with news articles about controversial topics. In these cases, the subjective pieces of information (people's opinions, politician's views, etc.) are likely more important than objective statements related to the topic. For instance, given a query "*partial birth abortion ban*", an opinionated sentence such as "*Eventually, he'd like all abortions to be banned because he believes they are murder*" is likely more important than another sentence such as "*We are performing so-called partial-birth abortions as defined by Kansas*

law". Similarly, when have an information need, this need is usually related to a person, a location or an organization. Therefore, sentences that contain named entities may be highly relevant. For instance, given the query "*U.S. Embassy bombings in Africa, 1998*", the sentence "*Suspected bombs exploded outside the U.S. embassies in the Kenyan and Tanzanian capitals Friday [...]*" might be more important than another sentence such as "*At least 82 were killed and more than 1,700 injured, officials said as dawn broke Saturday*". Finally, the use of sentence length as a query-independent feature may also help because long sentences usually provide more information than short ones and, therefore, they are more likely relevant (some short sentences act solely as connectors between the pieces of the discourse).

Summing up, the set of features considered in our study are:

a) **opinion-based features**, including the subjectivity nature of sentences (a sentence may be objective or subjective) and the polarity of the sentence terms (the number of positive terms in a sentence, the number of negative terms, and the number of opinionated terms). Observe that sentence retrieval is an appropriate scenario to study these issues because sentences are compact pieces of information and their subjective or objective nature can be reasonably estimated (with coarse-grained chunks such as documents or paragraphs, this opinion-based classification is more problematic because it is hard to classify a document as subjective or objective).

b) **named entities features**, i.e. names of persons, locations, organizations, etc.

c) **sentence length**, i.e. the number of terms in a sentence, ignoring stopwords.

The features described above are considered in isolation or in combination. This helps to understand the configuration of query-independent features that performs the best. In order to incorporate these sentence features as query-independent evidence into SR models, we follow a formal methodology based on kernel density estimation [CRZT05]. We show that the combination of these query-independent features with state of the art SR scores yields to important improvements in performance with negligible computational costs at retrieval time.

1.4.1 Combining Content Match and Query-Independent Scores

As argued above, we use tfisf as our sentence retrieval baseline because it performs reasonably well and it is parameter-free. However, the performance of these state of the art SR models is still weak. This problem is especially aggravated when handling short queries because of the little overlap between queries and sentences. The vocabulary mismatch problem arises severely in SR and, therefore, the models should not be solely based on query-sentence matching scores. Thus, SR methods need to include additional evidence besides content match evidence. A natural way to address the problem consists of defining query-independent weights that modify sentence retrieval scores. To this aim we apply a Feature's Log Odds Estimation (FLOE), which is a formal methodology designed by Craswell et al [CRZT05]. FLOE is a density analysis method that models the transformation needed in order to add query-independent features into existing retrieval models. It is a formal and powerful method that suggests good functional forms to transform feature values into relevance scores, without assuming independence between the feature and the baseline. In web document retrieval, FLOE has been used to define transformations for BM25 in order to include features such as PageRank, indegree or URL length. FLOE is, therefore, a natural choice to combine tfisf with query-independent features in our SR scenario.

1.4.1.1 Score Adjustment Under Independence

The aim of the SR task is to rank sentences (S) according to the probability they are relevant (R), $p(R|S)$. The probability score in a log-odds way that preserves the rank order with respect to a query (Q) can be expressed as:

$$p(R|S) \overset{Q}{\propto} \log \frac{p(R|S)}{p(\bar{R}|S)} \overset{Q}{\propto} \log \frac{p(S|R)}{p(S|\bar{R})} \tag{1.8}$$

We can now consider that sentences have two components: a content match component (M) and a query-independent (or static score) component (I), which is related to a given query-independent feature (e.g. in web retrieval I might be the variable associated to the PageRank of the document). Given these two components, Equation 1.8 can be separated into two additive scores:

$$\log \frac{p(S|R)}{p(S|\bar{R})} = \log \frac{p(M,I|R)}{p(M,I|\bar{R})} = \log \frac{p(M|R)}{p(M|\bar{R})} + \log \frac{p(I|M,R)}{p(I|M,\bar{R})} \tag{1.9}$$

A standard matching function (e.g. tfisf) can play the role of the first addend [CRZT05]:

$$\log \frac{p(S|R)}{p(S|\bar{R})} \overset{Q}{\propto} \text{tfisf} + \log \frac{p(I|M,R)}{p(I|M,\bar{R})} \tag{1.10}$$

If components I and M were independent, then

$$\log \frac{p(I|M,R)}{p(I|M,\bar{R})} = \log \frac{p(I|R)}{p(I|\bar{R})} \tag{1.11}$$

and, therefore, the adjustment under this independence assumption would be:

$$indep(I,R) = \log \frac{p(I|R)}{p(I|\bar{R})} \tag{1.12}$$

Because the number of relevant sentences is very small compared to the number of sentences in the collection, the independence-based adjustment can be approximated as:

$$indep(I,R) = \log \frac{p(I|R)}{p(I)} \tag{1.13}$$

1.4.1.2 Feature's Logs-Odd Estimator

The adjustment described above would perform well if the baseline and the query-independent feature would be actually independent. However, it might be the case that the baseline already retrieve sentences with proper levels of the feature I. If so, the indep score would overstate the score boost of the feature (double-counting). The Feature's Logs-Odd Estimator (FLOE) is a method designed by Craswell et al [CRZT05] that avoids double counting by analyzing the levels of the feature in the baseline.

FLOE is a method that computes the probability estimates for a set of sentences. Given r, the number of known relevant sentences for a given query, FLOE takes the top r retrieved sentences from the baseline for each query and computes the probability estimates for this set as we describe next.

Let T bet the set of top r retrieved sentences and $\bar{T}$ be the remaining sentences in the collection. The estimate for this set is defined as:

$$\log \frac{p(I|T)}{p(I|\bar{T})} \approx \log \frac{p(I|T)}{p(I)} \tag{1.14}$$

This value represents the behavior of the baseline with respect to a given feature I. If we subtract this weight from indep, we obtain the part of the feature weight that is not captured by the baseline:

$$FLOE(I, R, T) = \log \frac{p(I|R)}{p(I)} - \log \frac{p(I|T)}{p(I)} = \log \frac{p(I|R)}{p(I|T)} \tag{1.15}$$

Therefore, FLOE corrects the behavior of the baseline to achieve the overall adjustment suggested by indep.

The adjustment defined by FLOE was not successful when used directly to combine BM25 and features such as PageRank, indegree, URL Length and Click Distance [CRZT05]. However, we will demonstrate in our work that FLOE "as is" helps directly to enhance significantly the SR performance.

1.4.2 Query-Independent Features

In this section we explain the query-independent sentence features used in our study.

1.4.2.1 Opinion-Based Features

We hypothesize that SR methods can be further improved by leading the retrieval process towards opinionated sentences.

Opinion mining (also known as Sentiment Analysis, Subjectivity Analysis, Review Mining or Appraisal Extraction) deals with computation treatment of opinions, sentiment and subjectivity in texts [PL08a]. This involves a number of challenging goals including opinion detection, identification of opinion holders and their authority, estimation of the polarity of the opinions, etc. The core component of these systems is usually a classifier, whose purpose is detecting opinions and estimating their polarity. Document passages (and sentences as a particular case) may be classified following their opinionated nature. For instance, sentences can be labeled as objective or subjective. Additionally, subjective material can be classified as expressing either an overall positive or an overall negative opinion. For instance, the sentence *"it is a way to call attention to the fact that he thinks the death penalty is offensive and obscene"* is a subjective sentence that has a negative connotation (about the death penalty). The research problems involved in these estimations are currently being addressed from different perspectives supported by a wide range of research areas [PL08a]. In 2006, 2007, 2008 and 2009 the TREC Blog Tracks [OdRM$^+$06, MOS07, OMS08, MOS09] were created to explore the information seeking behavior in the blogosphere. They

are a standard benchmark to help researchers in designing new efficient and effective opinion retrieval algorithms.

Our intuition is that users tend to be mostly interested in subjective information, especially when they look for news articles about controversial topics. In such cases, the subjective information is likely more important than objective statements related to the topic. We think that, by extracting opinionated information in sentences, we could improve the estimation of sentences' relevance.

Opinion-based features associated to every sentence were extracted by using a highly effective opinion mining software named OpinionFinder [WR05]. OpinionFinder is a state of the art subjectivity detection system [PL08a, PL08b] that processes texts and labels sentences (and parts of sentences) following their subjectivity and polarity nature. The text is first processed using part-of-speech tagging, name entity recognition, tokenization, stemming and sentence splitting. Next, using a dictionary-based method, a parsing module builds dependency parse trees where subjective expressions are identified. This is powered by Naive Bayes classifiers that are trained on sentences automatically generated from unannotated data. These classifiers have been shown to perform very well with several opinion corpus [WR05].

OpinionFinder classifies sentences as subjective or objective (or unknown if it cannot determine the nature of the sentence). Subjective sentences express private states, which are internal, mental or emotional states, including speculations, beliefs, emotions, evaluations, goals, and judgments (e.g. *"Peter thought he won the championship"* or *"Anne hoped her meeting would go well"*). In this respect, OpinionFinder implements two classifiers: an accuracy classifier and a precision classifier. The accuracy classifier yields the highest overall accuracy. It tags each sentence as either subjective or objective. The precision classifier optimizes precision at the expense of recall. It classifies a sentence as subjective or objective only if it can do so with confidence (otherwise, it tags the sentence as unknown). Additionally, OpinionFinder identifies the polarity of sentence terms, i.e. tags terms that are estimated to express positive or negative feelings (e.g. *"hope"* is a term with positive polarity but *"disaster"* is a term with negative polarity).

In this paper we work with the following set of opinion-based features: the subjective nature of the sentence (F_{subj}), which is a binary value (1 when the sentence is classified as subjective and 0 otherwise); the number of positive terms in a sentence (F_{pos}); the number of negative terms in a sentence (F_{neg}); and the number of opinionated terms in a sentence (F_{opt}), i.e. the number of either positive or negative terms.

Indep and FLOE Adjustments

Now, we analyze the indep and FLOE adjustments considering each of

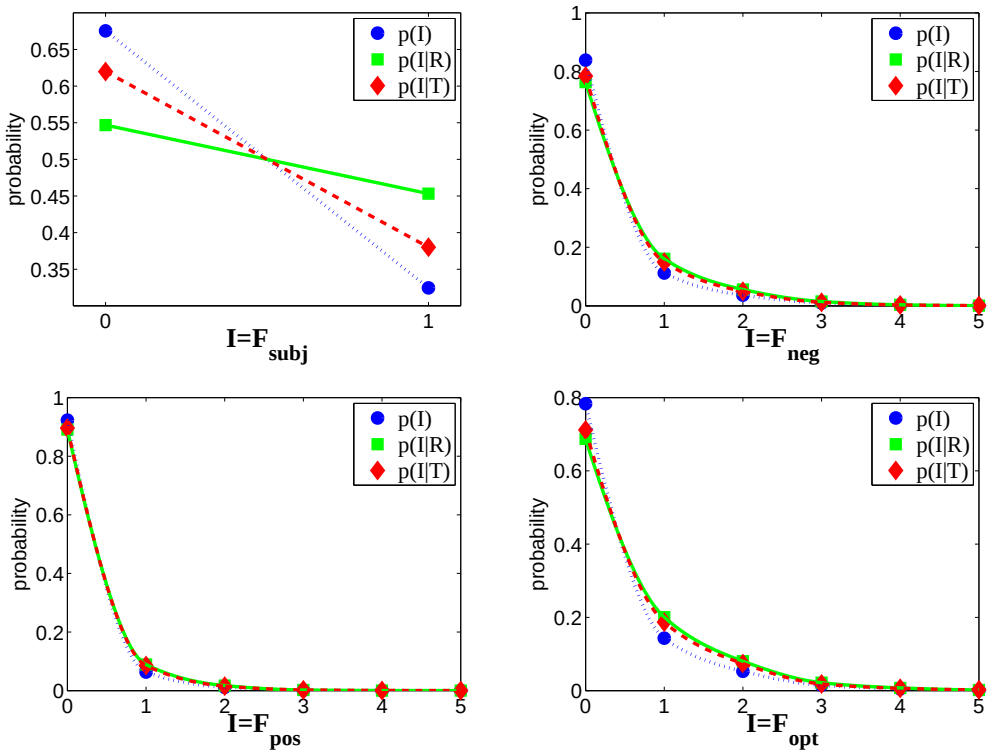


Figure 1.5: $p(I)$, $p(I|R)$ and $p(I|T)$ for the opinion-based features.

the opinion-based features explained above.

In Figure 1.5 we show the curves for the probabilities $p(I|R)$, $p(I|T)$ and $p(I)$ needed to compute indep and FLOE, where I is an opinion-based feature. We show the graphs considering one of our training collections[7] (the TREC 2003 Novelty Track dataset [SH03]). With the features F_{neg}, F_{pos} and F_{opt} (polarity features), the graphs were smoothed by applying a shape preserving interpolation[8].

The probabilities shown in Figure 1.5 let us predict the behavior of indep and FLOE adjustments. With F_{subj}[9], $p(I = 0) > p(I = 0|R)$ and $p(I = 1) < p(I = 1|R)$, meaning that the percentage of subjective sentences in the relevant set is higher than the overall percentage of subjective sen-

[7]Similar plots were obtained for all training collections described in Section 1.4.3.

[8]We used the shape preserving interpolation instead of kernel density smoothing (used by Craswell et al [CRZT05]) because our features are discrete. Our curves were smoothed to show more clearly the evolution of the feature's values. However, the features take integer values and, therefore, the graphs should only be analyzed on those points whose x coordinate takes an integer value.

[9]In the plot we only consider subjectivity given the accuracy classifier.

tences in the entire collection. This supports our hypothesis that promoting subjective sentences might lead to higher performance because the distribution of subjective sentences is comparably larger in the set of relevant sentences than in the collection. On the other hand, $p(I = 0|T) > p(I = 0|R)$ and $p(I = 1|T) < p(I = 1|R)$ indicate that the SR baseline does not retrieve enough subjective sentences and, therefore, promoting subjective sentences should improve performance. Moreover, $p(I = 0|T) < p(I = 0)$ and $p(I = 1|T) > p(I = 1)$. This means that the SR baseline retrieves some subjective material and, therefore, the independence assumption is not completely satisfied for this feature.

It is harder to find clear trends with the polarity features (F_{neg}, F_{pos} and F_{opt}). Still, observe the following slight tendency: $p(I = 0) > p(I = 0|R)$ and $p(I = n) < p(I = n|R)$ (with $n > 0$) for any of the polarity features. This means that the percentage of sentences that contain any of the defined polarity terms is higher in the relevant set than in the collection and, therefore, these polarity features may help to estimate the relevance of sentences. However, note that, in these cases, $p(I|R)$ is close to $p(I|T)$. This indicates that the SR baseline retrieves many sentences with proper levels of the feature and, likely, these features will not help to estimate the relevance of sentences as much as F_{subj} will do. Additionally, with F_{pos}, the distinction between $p(I)$ and $p(I|R)$ is blurred, and $p(I|T) \approx p(I|R)$. We therefore anticipate that positive terms will be a less valuable indicator of relevance and, because F_{opt} is the sum of F_{pos} and F_{neg}, results obtained with the F_{opt} will be similar to the ones obtained with F_{neg}.

The indep score represents the adjustment suggested under the independence assumption, i.e. baseline and features are independent. In Figure 1.6 we show the indep curves. For any of the opinion-based features explained above, indep suggests assigning more weight to sentences with opinionated material ($F_{subj} = 1$ or F_{pos}, F_{neg}, $F_{opt} \geq 1$) and even remove some weight to those sentences that do not contain any opinionated information (e.g. $F_{subj} = 0$ leads to a negative weight).

FLOE corrects the behavior of the baseline to achieve the overall adjustment suggested by indep. In Figure 1.7 we represent graphically $\log \frac{p(I|T)}{p(I)}$, indep and the FLOE adjustment. Because the independence assumption does not hold, indep and FLOE adjustments are different. However, trends are similar, in general, for both cases: increasing relevance scores to those sentences containing opinionated information and decreasing scores to the rest of sentences. Note that, in the case of F_{pos}, the FLOE adjustment is erratic (as argued above, we do not expect any benefit from adjustments based on F_{pos}).

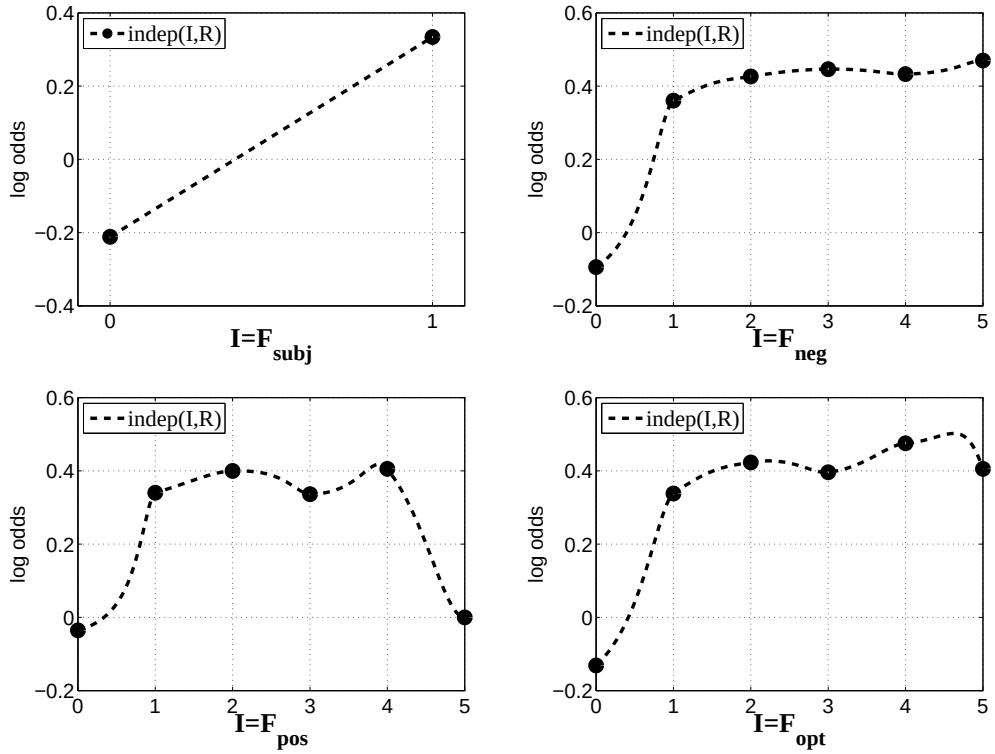


Figure 1.6: Adjustment under independence assumption, given the opinion-based features.

1.4.2.2 Features Based on Named Entities

Named entities (NEs) are proper names, such as names of persons, locations, organizations, etc. We hypothesize that the introduction of named entities into existing sentence retrieval models may help to estimate the relevance of sentences. These named entities are usually core components of a given text and user information needs might depend strongly on people names (*"How many albums does Madonna have?"*), locations (*"Olympic Games in Barcelona"*) or organizations (*"Microsoft antitrust charges"*). Hence, in this subsection we study different named entities as query-independent features and incorporate them into existing sentence retrieval methods by applying FLOE. The use of these name entities as query-independent features was considered in [LC08]. However, their work was focused on using these features for the novelty detection task.

Name Entity Recognition (also known as entity identification or entity extraction) is a subtask of Information Extraction consisting of identifying atomic parts of text and classify them into name categories, such as name of persons (e.g. *"Mary"*, *"Richard"*), locations (e.g. *"New York"*, *"Spain"*,

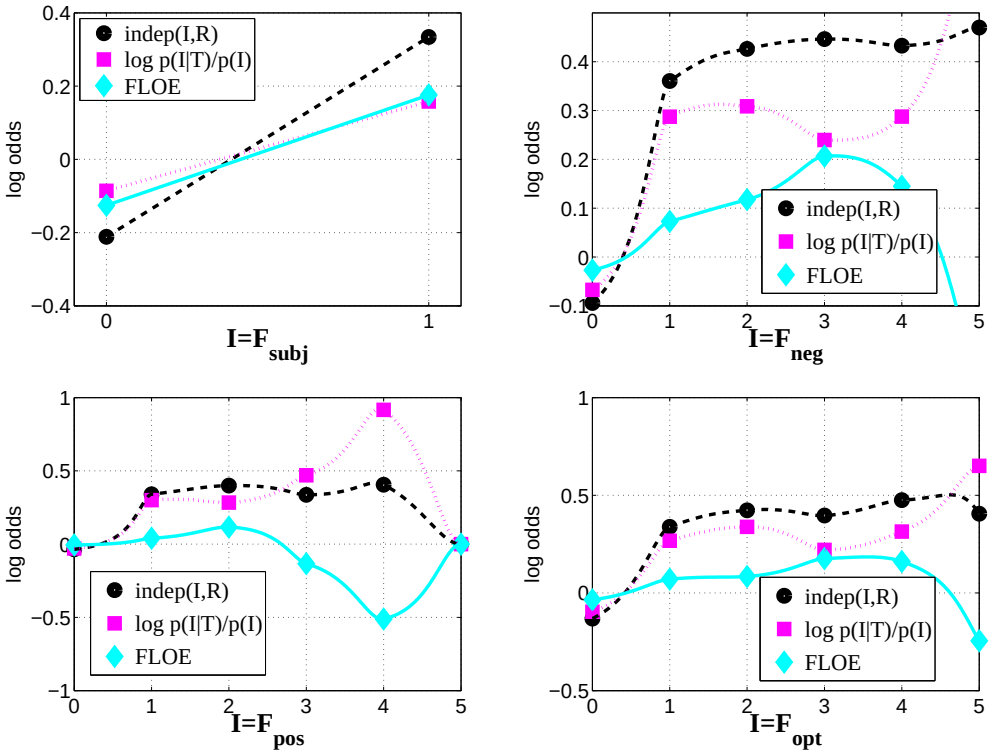


Figure 1.7: FLOE adjustment for the four opinion-based features. Adding FLOE to log $\frac{p(I|T)}{p(I)}$ we get the ideal adjustment, indep.

"Europe"), organizations (e.g. *"World Health Organization"*, *"Microsoft"*), expressions of time (*"yesterday"*, *"five years ago"*, *"in 1951"*), etc. In order to identify these named entities we utilize the CRFClassifier [FGM05], an implementation of the linear chain Conditional Random Field sequence model (a framework for building probabilistic models to segment and label sequence data [LMP01]) provided by the University of Standford. This software identifies atomic sequences of words in a text which are names of things (persons, locations and organizations names) and classifies them as names of entities.

In this work, different named entity evidences are considered as query-independent features: the number of person's names in a sentence (F_{pers}), the number of location's names in a sentence (F_{loc}), the number organization's names in a sentence (F_{org}), and the overall number of named entities (person, location and organization names) in a sentence (F_{ne}).

Indep and FLOE Adjustments

Given the NE features explained above, we analyze now the indep and FLOE adjustments.

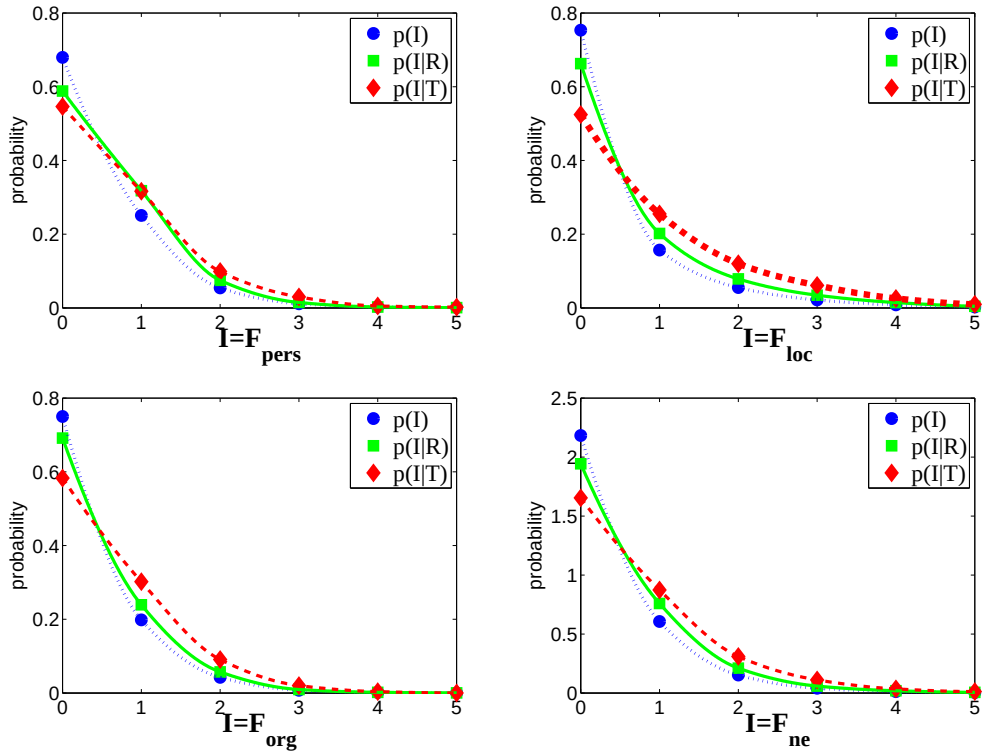


Figure 1.8: $p(I)$, $p(I|R)$ and $p(I|T)$ for the NE features.

In Figure 1.8 we show the smoothed curves[10] for the probabilities $p(I|R)$, $p(I|T)$ and $p(I)$ we need to estimate indep and FLOE, where I is one of the NE features. Again, we show here the graphs given only one of our training collections (TREC 2003) because trends are similar in any of the collections.

Given any of the NE features studied here, the trends shown in Figure 1.8 are similar: $p(I = 0) > p(I = 0|R)$ and $p(I) \leq p(I|R)$ with $I \geq 1$. This means that the percentage of sentences that contain named entities is higher in the relevance set than in the collection. This supports our belief that promoting sentences with named entities might be a way to improve performance. Additionally, $p(I = 0|T) < p(I = 0)$ and $p(I|T) > p(I)$ with $I \geq 1$, meaning that the SR baseline already retrieves sentences containing named entities (on average, there are more sentences with named entities in the retrieved set compared with the collection as a whole).

In Figure 1.9 we show the adjustment suggested under the independence assumption. Given the feature F_{pers}, the indep adjustment suggests to increase the weight to sentences with at least a person name, but where value of

[10] Again, we smoothed these curves by applying a shape preserving interpolation.

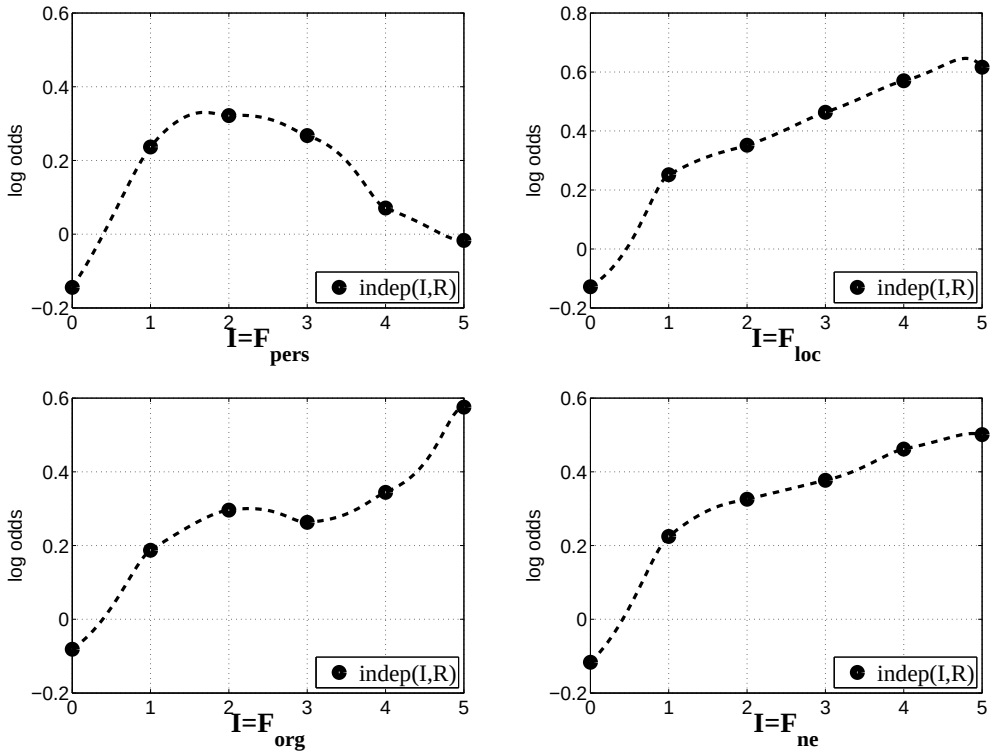


Figure 1.9: Adjustment under independence assumption given the features based on named entities.

F_{pers} is not too high ($F_{pers} \geq 1$ and $F_{pers} < 5$), and to remove some weight to sentences with no person names. For the remaining features, the adjustment suggests to increase the weight of sentences that contain at least one named entity (F_{loc}, F_{org} and $F_{ne} \geq 1$) and to remove some weight to sentences with no named entities.

However, the independence assumption does not hold here either because the baseline already retrieves sentences that contain named entities. In Figure 1.10 we show the FLOE adjustment given the NE features. Unlike indep, FLOE adjustments suggest, in general, to remove some weight to sentences that contain at least a named entity (except in the cases $F_{pers} = 1$ and $F_{org} \geq 5$) and to increase the weights to sentences with no named entities. These trends are the opposite as indep because, as shown in Figure 1.8, the proportion of sentences that contain named entities in the retrieved set is higher than in the relevance set ($p(I = 0|R) > p(I = 0|T)$ and $p(I|R) \leq p(I|T)$, with $I \geq 1$). Still, these trends reflect that these NE features might help to correct the behavior of the SR baseline.

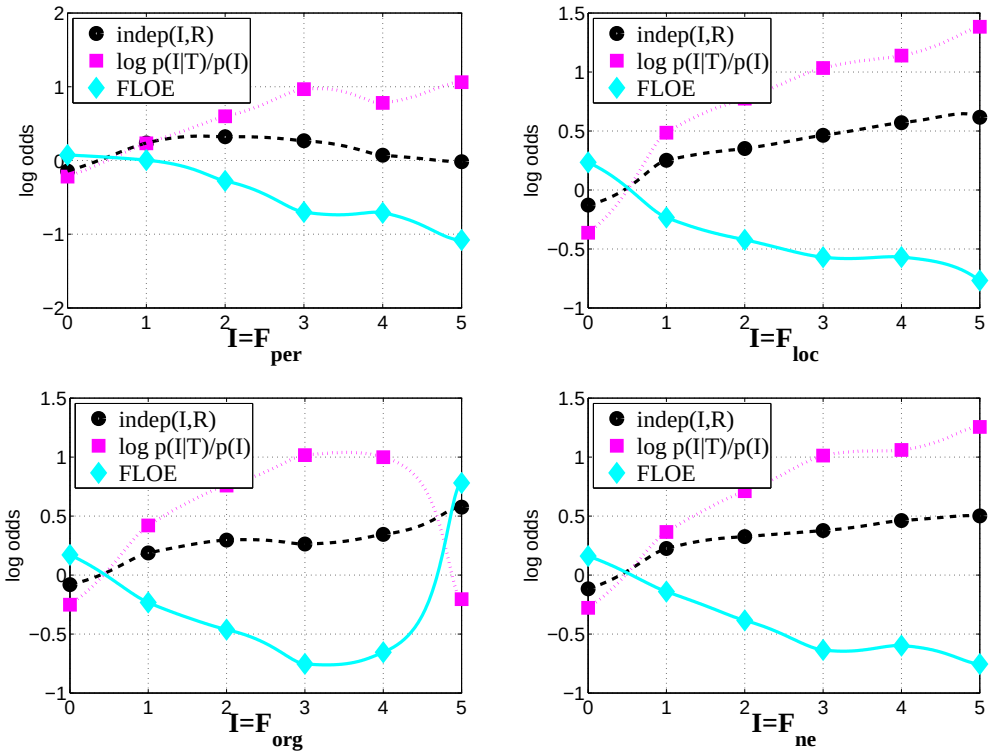


Figure 1.10: FLOE adjustment for the four NE features. Adding FLOE to $\log \frac{p(I|T)}{p(I)}$ we get the ideal adjustment, indep.

1.4.2.3 Sentence Length Feature

In document retrieval, document length has been recurrently used by a number of length retrieval normalizations, leading to highly effective retrieval mechanisms. The length of the retrieval unit, i.e. sentence length (F_{len}[11]), is, therefore, an important feature to be explored at sentence level. Observe also that regular length corrections, such as those implemented by BM25, do not work well in SR. In fact, the optimal performance of BM25 (Table 1.2) was found with $b = 0$ (i.e. no length correction). This means that standard length corrections are not well suited to SR problems. However, FLOE could suggest alternative length normalizations that work properly in SR. In this respect, we study here the impact of sentence length as a new feature.

Indep and FLOE Adjustments

In Figure 1.11 we show the curves of the probabilities $p(I|R)$, $p(I|T)$ and $p(I)$ needed to compute indep and FLOE, where I is the sentence length

[11]Sentence length refers here to the number of words in a sentence, ignoring stopwords (the list of the stopwords we used is available in Appendix A).

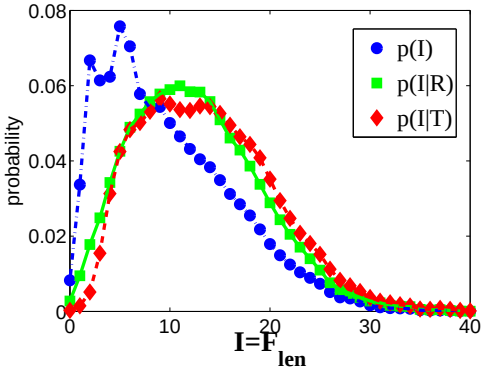


Figure 1.11: $p(I)$, $p(I|R)$ and $p(I|T)$ for F_{len}.

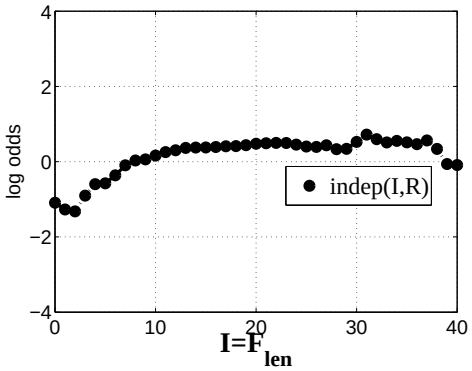


Figure 1.12: Adjustment under independence assumption given F_{len}.

feature (F_{len}). In this case, $p(I) > p(I|R)$ with $0 \leq I < 8$ and $p(I) > p(I|R)$ with $I \geq 8$. This means that the proportion of relevant sentences tends to contain longer sentences on average compared to the collection. On the other hand, $p(I|R) \geq P(I|T)$, with $0 \leq I < 15$ and $p(I|R) < p(I|T)$ with $I \geq 15$. This means that our SR baseline is retrieving sentences that are longer than the average sentences in the collection and, therefore, it looks like there is a good match between the retrieval and relevance patterns.

In Figure 1.12 we show the adjustment under the independence assumption, and in Figure 1.13 we show $\log \frac{p(I|T)}{p(I)}$, indep and the FLOE adjustment. The indep adjustment suggests to increase the weight to those sentences that contain at least 9 terms ($F_{len} \geq 9$) and to reduce the weight to shorter sentences. In contrast, FLOE suggests to remove weight to sentences longer than 15 and giving more weight to sentences that contain less than 15 terms.

In the next subsection we show the performance of the SR baseline after applying these adjustments, given the different features.

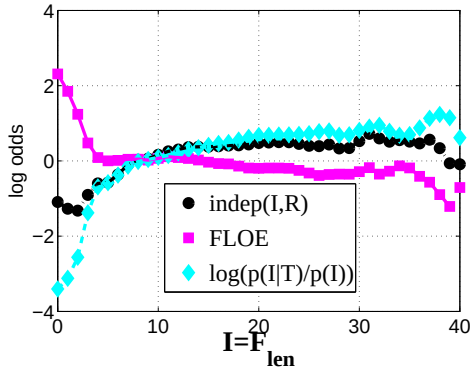


Figure 1.13: FLOE adjustment for F_{len}. Adding FLOE to $\log \frac{p(I|T)}{p(I)}$ we get the ideal adjustment, indep

1.4.3 Experiments

We tested the performance of our variants considering the TREC 2003 and TREC 2004 datasets. TREC 2002 was not considered, however, because only the 2% of sentences were estimated as relevant and, therefore, given this marginal amount of relevant sentences, this collection is not appropriate for statistical estimation. The statistics constructed upon this data would not be reliable. As a matter of fact, the characteristics of the TREC 2002 Novelty Track data have been criticized in the past and TREC 2003 and TREC 2004 are regarded as more robust sentence retrieval and novelty benchmarks [Li06].

In TREC 2003 and TREC 2004 topics are of two classes: events (e.g. "*Find details about the bombing at the 1996 Olympics in Atlanta*") or opinions (e.g. "*Find world-wide opinions about the death penalty*"). We can therefore test our methods with an assorted set of information needs. This helps to understand when opinion-based features are useful (e.g. are these features only useful for information needs that explicitly demand opinions?). In this respect, we report, first, results obtained with the complete set of topics and, next, we discuss the relative effects on different types of topics.

1.4.3.1 Experiments with Opinion-Based Features

We now test the incorporation of opinion-based features into tfisf. The training stage consists of applying FLOE (Equation 1.15) in the training collection to obtain query-independent adjustments such as those shown in Figure 1.7. Next, these adjustments are applied in the test collection. Given a sentence

		tfisf+FLOE				
	tfisf (baseline)	F_{subj} **(accuracy classifier)**	F_{subj} **(precision classifier)**	F_{neg}	F_{pos}	F_{opt}
	test: TREC 2003 (train: TREC 2004)					
P@10	.7480	**.7560**	.7520	.7580	.7320	.7520
Δ%		*(+1.07)*	*(+0.53)*	*(+1.34)*	*(−2.14)*	*(+0.53)*
MAP	.3851	**.3986*†**	.3892	.3899	.3800	.3896
Δ%		*(+3.51)*	*(+1.06)*	*(+1.25)*	*(−1.32)*	*(+1.17)*
	test: TREC 2004 (train: TREC 2003)					
P@10	.4300	**.4880*†**	.4560	.4500	.4420	.4440
Δ%		*(+13.49)*	*(+6.05)*	*(+4.65)*	*(+2.79)*	*(+3.26)*
MAP	.2358	**.2500*†**	.2414	.2441*	.2383	.2427*
Δ%		*(+6.02)*	*(+2.37)*	*(+3.52)*	*(+1.06)*	*(+2.93)*

Table 1.3: Retrieval performance of tfisf and tfisf+FLOE in the test collections given the opinion-based features.

S and a query Q, the final similarity associated to a sentence is[12]:

$$\text{tfisf(S,Q)} + FLOE(I, R, T_{tfisf}) \tag{1.16}$$

where R is the set of relevant sentences, I is the feature, and T is the top r ranked sentences retrieved by our SR baseline (tfisf) - r is the number of actual relevant sentences for Q.

Table 1.3 and Figure 1.14 report the performance of this method for the test collections given the opinion-based features. The best results are bolded in the Table.

The adjustment modeled by FLOE leads to improvements in performance and most of them are statistically significant. This is a remarkable achievement because FLOE was unable to produce directly (i.e. without further adjustments) significant improvements for document retrieval [CRZT05].

The results can be summarized as follows. First, the number of positive terms in a sentence, F_{pos}, does not lead to statistical significant improvements. As argued in Section 1.4.2.1, we already expected this outcome for F_{pos} because there is not a clear distinction between $p(F_{pos})$ and $p(F_{pos}|R)$. Second, the models incorporating the F_{neg} and F_{opt} features outperform clearly the baseline with both performance measures but the improvements are only statistically significant with MAP in TREC 2004. Third, F_{subj} appears to be the strongest feature. Detecting subjective sentences with the

[12]In the following, we use the notation T_{model} to clarify the *model* used to obtain the retrieved set of sentences.

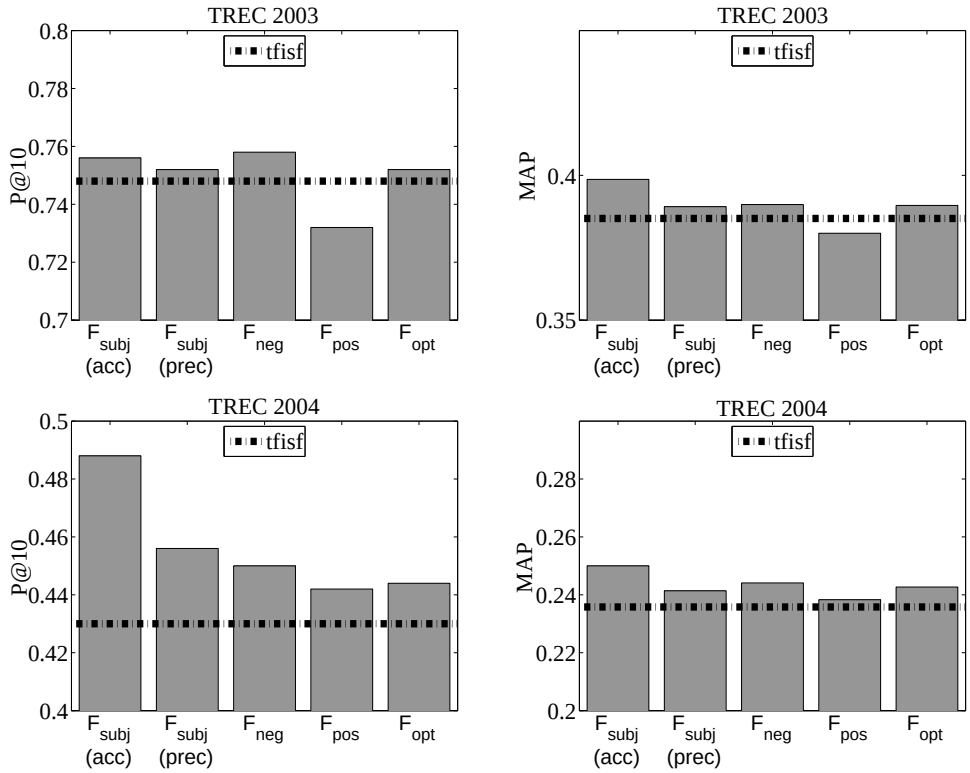


Figure 1.14: P@10 and MAP of tfisf and tfisf+FLOE in the test collections given the opinion-based features.

accuracy-based classifier, which is less stringent than the precision-based classifier, and including this evidence into the sentence retrieval model leads to very significant improvements in both P@10 and MAP.

Other Functional Forms Inspired by FLOE

In the previous section we showed that opinion-based features help significantly to retrieve relevant sentences. This positive outcome motivated us to go further and test other functional forms inspired by FLOE. Observe that the adjustments suggested by FLOE (e.g. Figure 1.7) might be less trustworthy in the regions of the plot with fewer examples. For instance, the number of sentences with more than four opinionated terms is much smaller than the number of sentences with one or two opinionated terms. This means that the right-hand end of the F_{neg}, F_{pos} and F_{opt} plots might be misleading. Note also that the forms of the FLOE curves can be easily approximated by simple functions such as lines. These functional forms might generalize better than the original FLOE adjustment and, therefore, they would avoid overfitting. We therefore propose in this section other alternatives to modify the

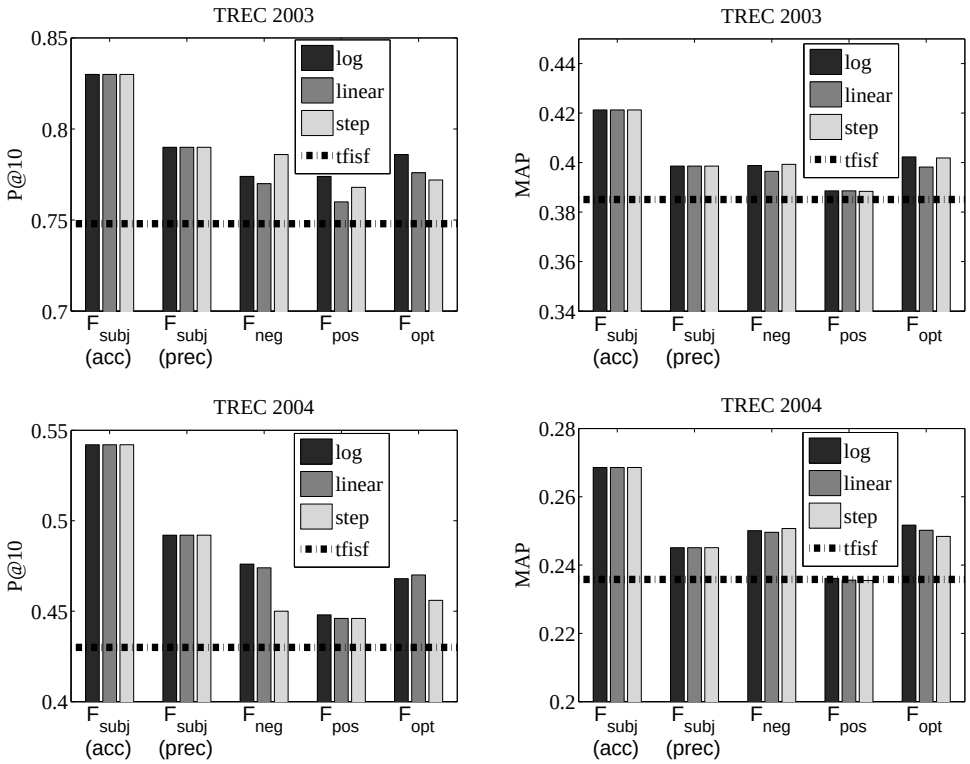


Figure 1.15: P@10 and MAP of tfisf and tfisf+function(I) in the test collections given the opinion-based features.

relevance weight with opinion-based evidence. Given a query-independent feature I, we tested the following functions:

$$log(I) = w \cdot \log(I + 1) \tag{1.17}$$

$$linear(I) = w \cdot I \tag{1.18}$$

$$step(I) = \begin{cases} 0 & \text{, if } I = 0 \\ w & \text{, otherwise} \end{cases} \tag{1.19}$$

where w is a weight that will be tuned in the training stage. The training stage for these new adjustments consists simply of tuning w to optimize performance[13]. The test results are shown in Table 1.4 and Figure 1.15. For F_{subj} we report only the linear function's results because this feature is binary and, therefore, all methods are virtually equivalent.

[13]We tested w with values ranging from 0 to 10 in steps of 0.1.

			tfisf+function(I)				
		tfisf (baseline)	F_{subj} (accuracy classifier)	F_{subj} (precision classifier)	F_{neg}	F_{pos}	F_{opt}
		test: TREC 2003 (train: TREC 2004)					
	log				.7740	.7740	.7860
	Δ%				(+3.48)	(+3.48)	(+5.08)
	w				2.3	2.5	2.4
	linear	.7480	**.8300***†	.7900	.7700	.7600*	.7760
P@10	Δ%		(+10.96)	(+5.61)	(+2.94)	(+1.60)	(+3.74)
	w		5.7	3.2	0.3	0.7	0.1
	step				.7860	.7680	.7720
	Δ%				(+5.08)	(+2.67)	(+3.21)
	w				2.8	3.3	6.0
	log				.3988*†	.3886*†	.4023*†
	Δ%				(+3.56)	(+0.91)	(+4.47)
	w				0.6	4.0	1.2
	linear	.3851	**.4213***†	.3986*†	.3965*†	.3886*†	.3982*†
MAP	Δ%		(+9.40)	(+3.51)	(+2.96)	(+0.91)	(+3.40)
	w		6.3	2.8	0.4	1.6	0.2
	step				.3993*†	.3884*†	.4019*†
	Δ%				(+3.69)	(+0.86)	(+4.36)
	w				0.4	3.3	2.8
		test: TREC 2004 (train: TREC 2003)					
	log				.4760	.4480	.4680*
	Δ%				(+10.70)	(+4.19)	(+8.84)
	w				1.6	3.4	0.4
	linear	.4300	**.5420***†	.4920	.4740*	.4460	.4700
P@10	Δ%		(+26.05)	(+14.42)	(+10.23)	(+3.72)	(+9.30)
	w		3.3	5.5	1.1	1.1	0.2
	step				.4500	.4460	.4560
	Δ%				(+4.65)	(+3.72)	(+6.05)
	w				1.1	0.9	0.9
	log				.2501*†	.2361	.2517*†
	Δ%				(+6.06)	(+0.13)	(+6.74)
	w				2.1	3.2	1.1
	linear	.2358	**.2686***†	.2451*†	.2496*†	.2356	.2502*†
MAP	Δ%		(+13.91)	(+3.94)	(+5.85)	(−0.08)	(+6.11)
	w		4.7	3.4	1.4	2.2	0.2
	step				.2507*†	.2355	.2484*†
	Δ%				(+6.32)	(−0.13)	(+5.34)
	w				2.5	3.3	3.0

Table 1.4: Retrieval performance of tfisf and tfisf+function(I) in the test collections given the opinion-based features.

The relative merits of F_{subj}, F_{neg}, F_{pos} and F_{opt} remain the same: F_{subj} is the strongest feature while F_{pos} is the weakest feature. The new adjustments

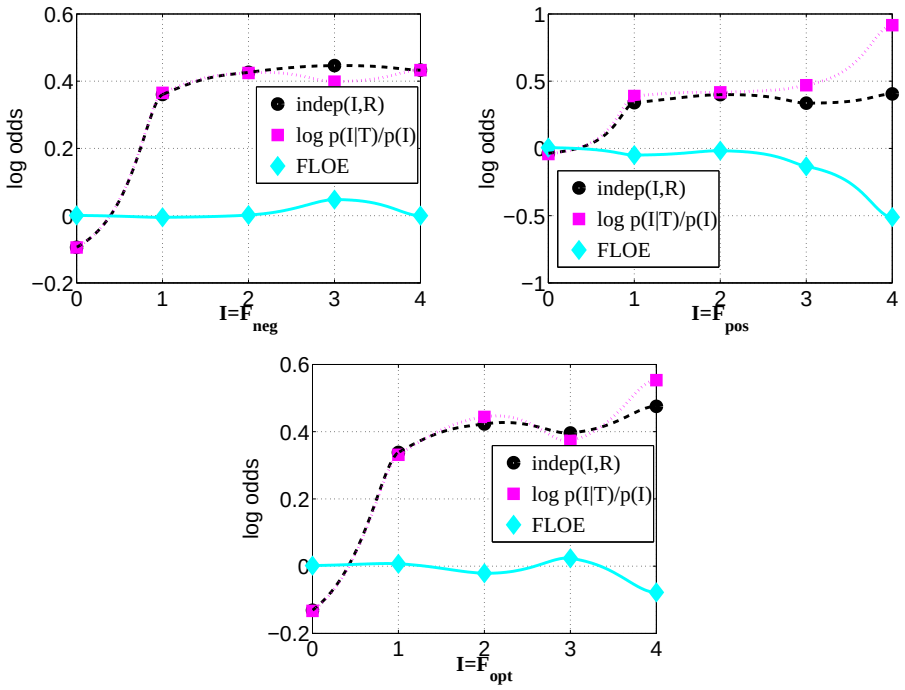


Figure 1.16: FLOE adjustment when combining F_{subj} with F_{neg}, F_{pos} and F_{opt}, respectively.

perform clearly better than the original FLOE's adjustment. Overall, there is no major difference between the functional forms tested but, in terms of statistical significance, the step function looks slightly worse than the others.

The experiments reported so far demonstrate that opinion-based features are important components that should not be disregarded when retrieving sentences. As a matter of fact, the performance of a state of the art sentence retrieval model improves very significantly when opinion-based features are included (e.g. F_{subj} leads to 9-26% improvements).

Post-Combination Checking

Having shown that these opinion-based features can individually produce benefits in terms of performance, it is natural to consider their combination. Rather than approaching this in an ad-hoc way, we resort again to FLOE. After adding a given feature, FLOE can predict whether or not another feature is still useful. Since F_{subj} is the feature that yields the highest precision, we took the strongest model designed so far (column in bold in Table 1.4, tfisf+linear(F_{subj})) and, given the top r sentences retrieved, we analyzed whether or not F_{neg}, F_{pos} and F_{opt} could be useful on top of these high performing approaches. This is shown in Figure 1.16.

Given a retrieved set produced by tfisf+linear(F_{subj}), the trends associated to the remaining features ($\log \frac{p(I|T)}{p(I)}$) match closely the ideal indep curve. This makes that the resulting FLOE's adjustment is flat around 0, indicating that no further adjustment is necessary. There is an exception for F_{pos} (it yields a decreasing pattern). Still, as shown above, this feature does not help to retrieve additional relevant material.

Discussion: Having demonstrated that opinion-based features help to retrieve relevant sentences, we analyze here the behavior of the sentence retrieval methods with different types of topics. As argued above, some TREC topics concern events while the remaining topics focus on opinions about controversial subjects such as cloning, gun control, and same-sex marriages. The test collection mentions explicitly the topic type (event or opinion) and, therefore, the performance results can be broken down by topic type.

In Table 1.5 and Figure 1.17 we compare the baseline and the strongest opinion-based model, tfisf+linear(F_{subj}), for both types of topics. The improvements achieved by the opinion-based model are consistent across query types. This clearly demonstrates that our models are very effective when handling opinion-oriented needs but they also lead to important improvements for event queries. For instance, given the event topic #N2 ("*Cloning of the sheep Dolly*"), the tfisf+linear(F_{subj}) model assigns high scores to relevant opinionated sentences such as "*[...] the scientists agree that the birth of the first clone sheep announced earlier this week is a major scientific breakthrough*". In contrast, this is not a top-ranked sentence with tfisf because this model takes only into account standard matching heuristics. This is a limitation because the proportion of subjective sentences in the relevant set is higher than the proportion of subjective sentences in the collection (see Figure 1.6). These results suggest that final users are particularly interested in subjective pieces of information regardless of the topic type. The few attempts done in the literature to apply subjectivity clues for sentence retrieval [KRH04, LC08] assumed that opinion-based methods are only effective for opinion topics. Our results demonstrate that this assumption is wrong.

The interest in subjective material might be either a particular feature of news datasets, such as most TREC collections, or a more general circumstance that holds in other domains. This will be subject to further research.

1.4.3.2 Experiments with Features Based on Named Entities

We test now the incorporation of features based on named entities into tfisf. In Table 1.6 and Figure 1.18 we show the performance in the test stage after incorporating these features (with the adjustment suggested by FLOE after

	events		**opinions**	
	tfisf (baseline)	**tfisf + linear (F_{subj})**	**tfisf (baseline)**	**tfisf + linear (F_{subj})**
	test: TREC 2003 (train: TREC 2004)			
	(#28 topics)		(#22 topics)	
P@10	.8143	.9071*†	.6636	.7318
Δ%		(+*11.40*)		(+*10.28*)
MAP	.4466	.4857*†	.3069	.3395*†
Δ%		(+*8.76*)		(+*10.62*)
	test: TREC 2004 (train: TREC 2003)			
	(#25 topics)		(#25 topics)	
P@10	.5240	.6000*	.3360	.4840*†
Δ%		(+*14.50*)		(+*44.05*)
MAP	.2770	.2953*†	.1947	.2420*†
Δ%		(+*6.61*)		(+*24.29*)

Table 1.5: Retrieval performance of tfisf and tfisf+linear(F_{subj}) considering event and opinion topics.

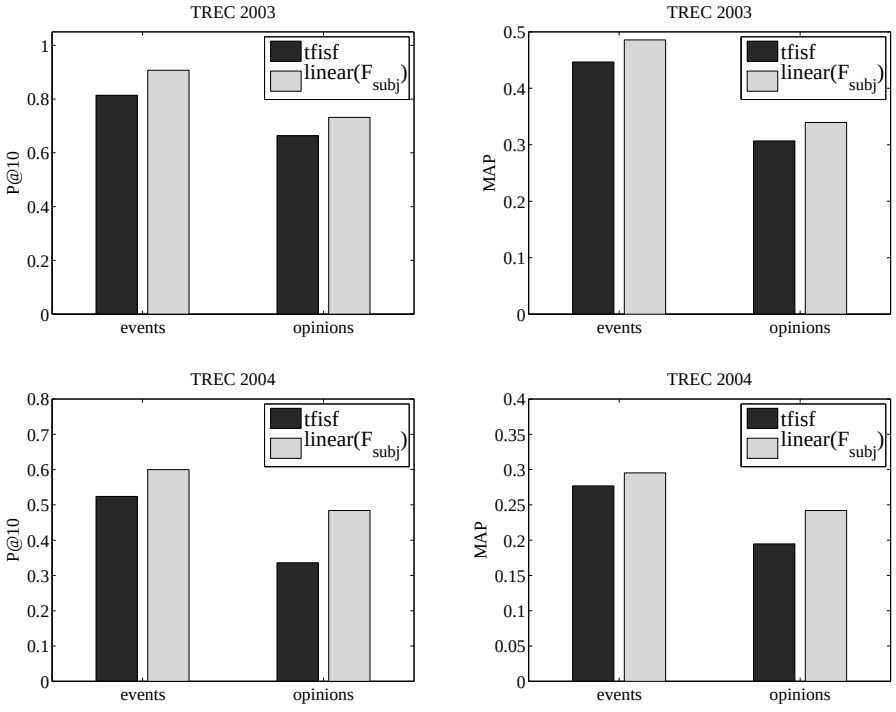


Figure 1.17: Retrieval performance of tfisf and tfisf+linear(F_{subj}) considering event and opinion topics.

		tfisf+FLOE			
	tfisf	$\boldsymbol{F_{pers}}$	$\boldsymbol{F_{loc}}$	$\boldsymbol{F_{org}}$	$\boldsymbol{F_{ne}}$
	test: TREC 2003 (train: TREC 2004)				
P@10	**.7480**	.7220*†	.7360	.7420	.7240*
$\Delta\%$		(-3.48)	(-1.60)	(-0.80)	(-3.21)
MAP	**.3851**	.3787*†	.3759*†	.3791*†	.3711*†
$\Delta\%$		(-1.66)	(-2.39)	(-1.56)	(-3.64)
	test: TREC 2004 (train: TREC 2003)				
P@10	.4300	**.4400**	.4100	**.4400**	.4360
$\Delta\%$		$(+2.33)$	(-4.65)	$(+2.33)$	$(+1.40)$
MAP	**.2358**	.2336	.2313*	.2319*	.2303*
$\Delta\%$		(-0.93)	(-1.91)	(-1.65)	(-2.33)

Table 1.6: Retrieval performance of tfisf and tfisf+FLOE in the test collections given the named entity-based features.

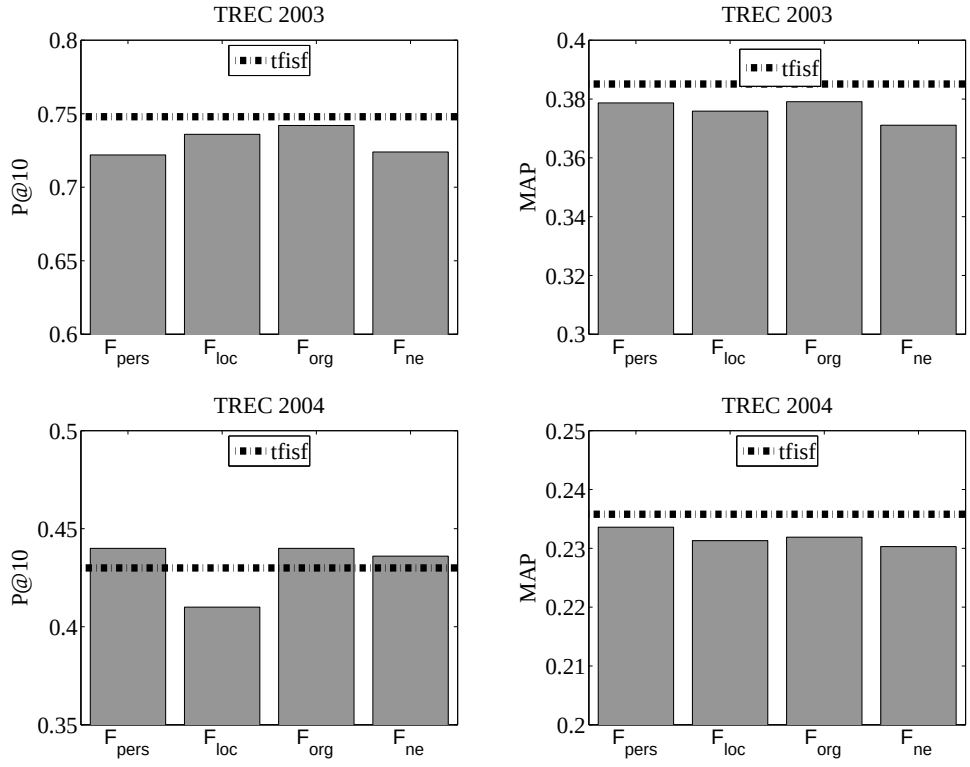


Figure 1.18: P@10 and MAP of tfisf and tfisf+FLOE in the test collections given the named entity-based features.

training).

In general, NE features do not help to improve performance. The adjust-

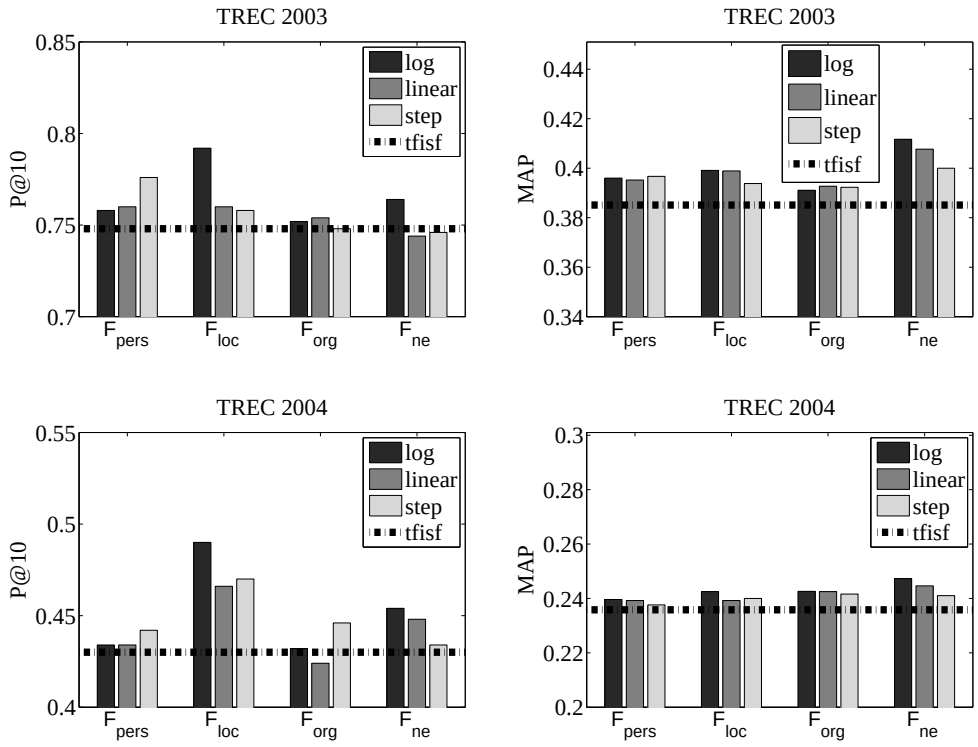


Figure 1.19: P@10 and MAP of tfisf and tfisf+function(I) in the test collections given the NE features.

ment suggested by FLOE is not beneficial here. There are few cases with improvements in performance but, anyway, those were not statistically significant. Anyway, we also applied empirical methods such as those tested for the opinion-based features. In Table 1.7 and Figure 1.19 we report the performance of these methods for the different NE features[14]:

In general, tfisf is not outperformed in terms of P@10. With MAP, F_{ne} is the feature that performs the best, regardless of the method (linear, log or step). F_{org} provides also good performance and F_{org} and F_{loc} only perform well with one of the collections.

Anyway, the improvements with respect to the baseline are usually modest and, usually, statistically insignificant. This may be happening because most queries contain named entities as explicit query terms. Therefore, the retrieval of sentences with named entities might be already guaranteed by the content match score (tfisf). In Table 1.8 we show the number of queries with and without named entities. There is a high number of queries con-

[14]We tested w values from -10 to 10, in steps of 0.1.

			tfisf+function(I)			
		tfisf	F_{pers}	F_{loc}	F_{org}	F_{ne}
		test: TREC 2003 (train: TREC 2004)				
	log		.7580	**.7920**	.7520	.7640
	Δ%		(+1.34)	(+5.88)	(+0.53)	(+2.14)
	w		0.4	1.4	0.2	1.2
P@10	**linear**	.7480	.7600	.7600	.7540	.7440
	Δ%		(+1.60)	(+1.60)	(+0.80)	(−0.53)
	w		0.1	0.1	0.3	0.2
	step		.7760	.7580	.7480	.7460
	Δ%		(+3.74)	(+1.34)	(+0.00)	(−0.27)
	w		0.9	1.7	0.6	1.4
	log		.3960*†	.3994*†	.3911	**.4117***†
	Δ%		(+2.83)	(+3.71)	(+1.56)	(+6.91)
	w		1.0	1.5	3.3	0.6
MAP	**linear**	.3851	.3952*†	.3989*†	.3927*	.4077*†
	Δ%		(+2.62)	(+3.58)	(+1.97)	(+5.87)
	w		0.4	0.6	0.7	2.6
	step		.3967*†	.3938*†	.3923*	.4000*†
	Δ%		(+3.01)	(+2.26)	(+1.87)	(+3.87)
	w		1.6	0.6	0.6	3.4
	test: TREC 2004 (train: TREC 2003)					
	log		.4340	**.4900***†	.4320	.4540
	Δ%		(+0.93)	(+13.95)	(+0.47)	(+5.58)
	w		2.3	1.4	1.7	3.6
P@10	**linear**	.4300	.4340	.4660	.4240	.4480
	Δ%		(+0.93)	(+8.37)	(−1.40)	(+4.19)
	w		0.5	1.2	0.9	0.6
	step		.4420	.4700*	.4460	.4340
	Δ%		(+2.79)	(+9.30)	(+3.72)	(+0.93)
	w		0.9	0.9	0.9	3.9
	log		.2396	.2425	.2426*†	**.2473***†
	Δ%		(+1.61)	(+2.84)	(+2.88)	(+4.88)
	w		2.5	4.3	1.5	4.3
MAP	**linear**	.2358	.2392	.2392	.2425*	.2446*
	Δ%		(+1.44)	(+1.44)	(+2.84)	(+3.73)
	w		0.9	2.5	1.0	1.5
	step		.2376	.2400	.2416*	.2410*
	Δ%		(+0.76)	(+1.78)	(+2.46)	(+2.21)
	w		3.0	3.0	2.0	3.0

Table 1.7: Retrieval performance of tfisf and tfisf+function(I) in the test collections given the named entity-based features.

taining named entities. Although FLOE still suggests some correction over the baseline (see e.g. Figure 1.7) it seems that this adjustment is not highly beneficial in terms of retrieval performance.

# named entities	F_{pers}	F_{loc}	F_{org}	F_{ne}
	TREC 2003			
0	35	35	41	21
≥1	15	15	9	29
	TREC 2004			
0	35	32	40	13
≥1	15	18	10	37

Table 1.8: Statistics of named entities per query.

1.4.3.3 Experiments with the Feature Based on Sentence Length

Sentence length was shown in the literature to be a helpful factor to improve standard document retrieval methods. We test here whether or not this also happens in the sentence retrieval scenario. Regular length corrections have been demonstrated to not work well in SR. For instance, BM25 performs clearly when the sentence length is ignored. However, FLOE could suggest alternative length normalizations that work properly in SR. Thus, in Table 1.9 and Figure 1.20 we show the performance of tfisf after applying the FLOE adjustment ($tfisf + FLOE$) and, additionally, the performance obtained after incorporating sentence length into tfisf with the empirical methods[15].

The performance of tfisf adjusted by FLOE does not outperform the standard tfisf method. With respect to the empirical methods, the log function performs better than the linear function. Improvements are statistically significant in most of the cases and, particularly, the benefits are more remarkable for MAP.

To sum up, we have improved our SR baseline by considering sentence length as a query-independent component. Although FLOE does not yield to improvements in performance over the baseline, we were able to outperform tfisf with an empirical method (linear or log). This might be because the FLOE adjustment for length (Figure 1.13) tends to be flat around 0 (for most of the length values) and, therefore, its effect is negligible. Furthermore, FLOE might be less trustworthly in the regions of the plot with a high number of terms (e.g. there are few sentences longer than 30 terms, as indicated by $p(I)$ plot in Figure 1.11) and, thus, empirical approximations might be more reliable in such situations.

[15]We do not show the performance of the step function because it makes no sense here (sentences contain always at least one term).

	tfisf	**tfisf + FLOE**	**tfisf + log**	**tfisf + linear**
	test: TREC 2003 (train: TREC 2004)			
P@10	.7480	.7300	.7640	**.7700**
$\Delta\%$		(-2.41)	$(+2.14)$	$(+2.94)$
w			2.0	0.1
MAP	.3851	.3749*†	**.4209***†	.4169*†
$\Delta\%$		(-2.65)	$(+9.30)$	$(+8.26)$
w			6.1	0.4
	test: TREC 2004 (train: TREC 2003)			
P@10	.4300	.4260	**.4680***	.4620
$\Delta\%$		(-0.93)	$(+8.84)$	$(+7.44)$
w			1.0	0.2
MAP	.2358	.2276*†	**.2622***†	.2596*†
$\Delta\%$		(-3.48)	$(+11.20)$	$(+10.09)$
w			5.6	0.3

Table 1.9: Retrieval performance of tfisf, tfisf+FLOE and tfisf+function(I) in the test collections given F_{len}.

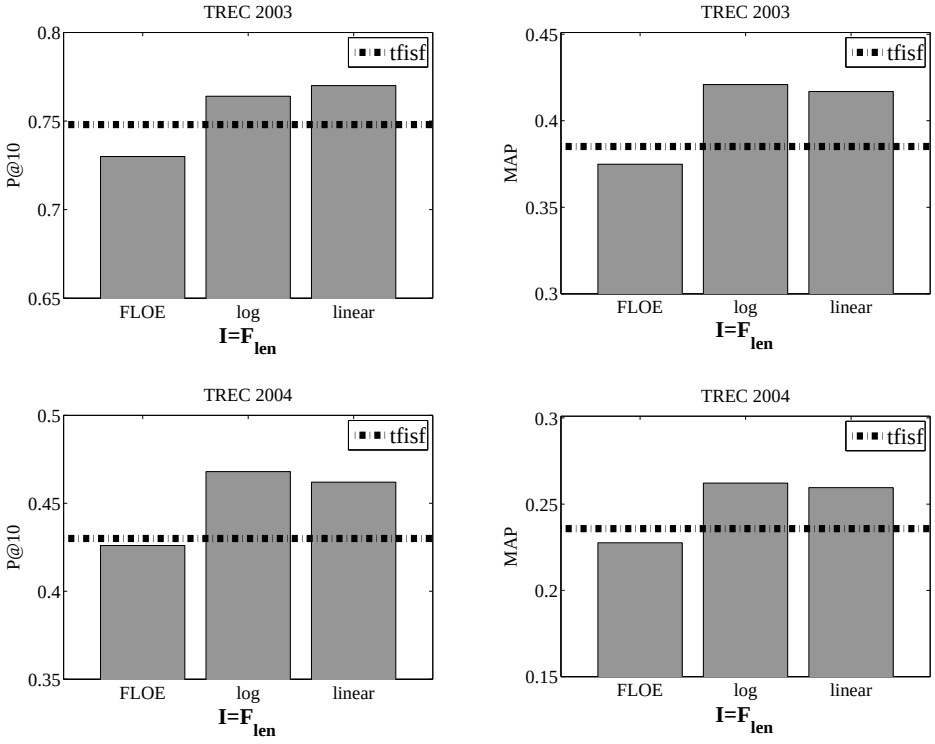


Figure 1.20: Retrieval performance of tfisf, tfisf+FLOE and tfisf+function(I) in the test collections given F_{len}.

	tfisf	tfisf + FLOE	tfisf + log	tfisf + linear
	test: TREC 2003 (train: TREC 2004)			
P@10	.7480	.8140*†	.8360*†	**.8380***†
$\Delta\%$		(+*8.82*)	(+*11.76*)	(+*12.03*)
w			0.4	0.1
MAP	.3851	.4122*†	**.4378***†	.4339*†
$\Delta\%$		(+*7.04*)	(+*13.68*)	(+*12.67*)
w			4.0	0.2
	test: TREC 2004 (train: TREC 2003)			
P@10	.4300	.5460*†	**.5580***†	.5440*†
$\Delta\%$		(+*26.98*)	(+*29.77*)	(+*26.51*)
w			0.1	0.1
MAP	.2358	.2613*†	**.2821***†	.2795*†
$\Delta\%$	(+*10.81*)	(+*19.64*)	(+*18.53*)	
w				

Table 1.10: Retrieval performance of tfisf, tfisf+FLOE and tfisf+function(I) in the test collections, combining subjectivity and sentence length features.

1.4.3.4 Combining Opinion and Sentence Length Features

In this subsection we explore the combination of different types of features. We have designed high performing sentence retrieval methods so far based on opinion estimation and sentence length. We will now assess the quality of combination following Craswell et al. [CRZT05] methodology. As a matter of fact, it is natural to wonder whether the proposed models can be further improved by combining features of different kind.

We do not consider here combinations involving named entity features because, as argued in Section 1.4.3.2, they lead to modest improvements in performance. Therefore, in order to study the combination of features, we only consider those combinations with higher impact on the estimation of sentence relevance: sentence length and opinion-based features. The approach followed here is to test sentence length on top of the best opinion-based sentence retrieval method, i.e. tfisf+linear(F_{subj}). First, we incorporate sentence length as a feature by using FLOE:

$$\text{tfisf}(S,Q) + \text{linear}(F_{subj}) + \text{FLOE}(len, R, T_{\text{tfisf+linear}(F_{subj})}) \tag{1.20}$$

As reported in Table 1.10 and Figure 1.21, after applying FLOE, improvements are statistically significant with respect to the SR baseline in terms of performance.

Besides FLOE, we also experimented with other functional forms such as linear and log transformations of the sentence length. Again, we do not

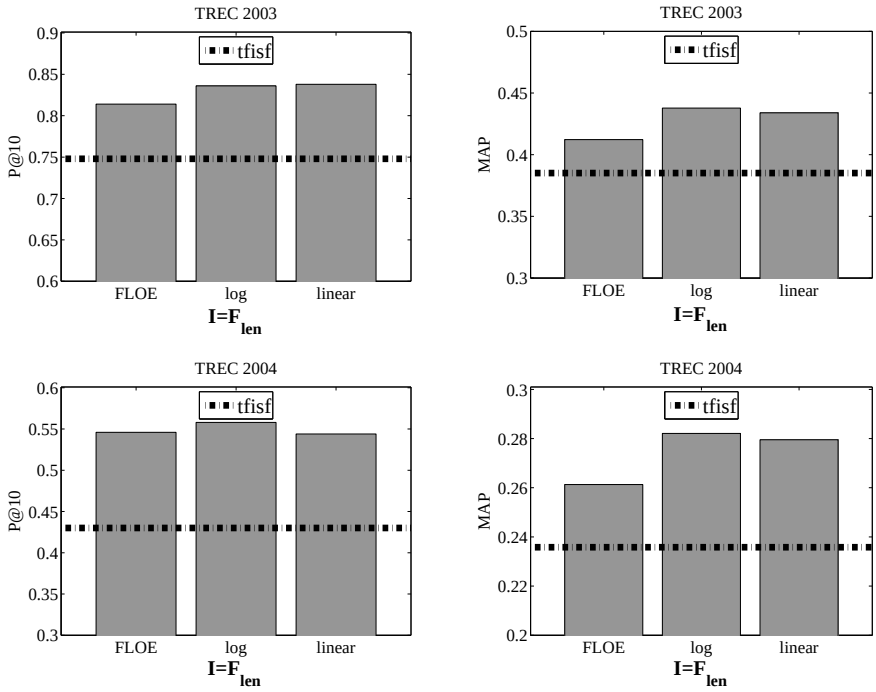


Figure 1.21: Retrieval performance of tfisf, tfisf+FLOE and tfisf+function(I) in the test collections, combining subjectivity and sentence length features.

apply the step function because sentences contain always at least a query term. The performance of these methods is also reported in Table 1.10 and Figure 1.21.

Empirical methods lead to significant improvements in performance. The combination of F_{subj} and sentence length (tfisf+linear(F_{subj})+log(F_{len})) leads to very effective sentence retrieval methods that outperform clearly the baseline. These results indicate that the effectiveness of sentence retrieval can be further improved not only by incorporating opinion-based features but also by combining them with sentence length weights. To further check that this combination is better than the model tfisf+linear(F_{subj}), Table 1.11 and Figure 1.22 report a comparison between the best opinion-based model (tfisf+linear(F_{subj})) and the best combination of opinions and length. This shows that, in terms of P@10, there is no need to include a sentence length factor. In contrast, a sentence length weight helps to improve performance in terms of MAP.

This evaluation demonstrates how powerful opinion-based features can be when used as *a priori* evidence for estimating the relevance of the sentences. The high performing baseline, tfisf, has been significantly enhanced

	tfisf+linear(F_{subj})	**tfisf+linear(F_{subj}) + log(F_{len})**
	test: TREC 2003 (train: TREC 2004)	
P@10	.8300	.8360
Δ%		(+0.72)
MAP	.4213	.4378*†
Δ%		(+3.92)
	test: TREC 2004 (train: TREC 2003)	
P@10	.5420	.5580
Δ%		(+2.95)
MAP	.2686	.2821*†
Δ%		(+5.03)

Table 1.11: Comparison between the best opinion-based model (tfisf+linear(F_{subj})) and the best combination model (tfisf+linear(F_{subj})+log(F_{len})).

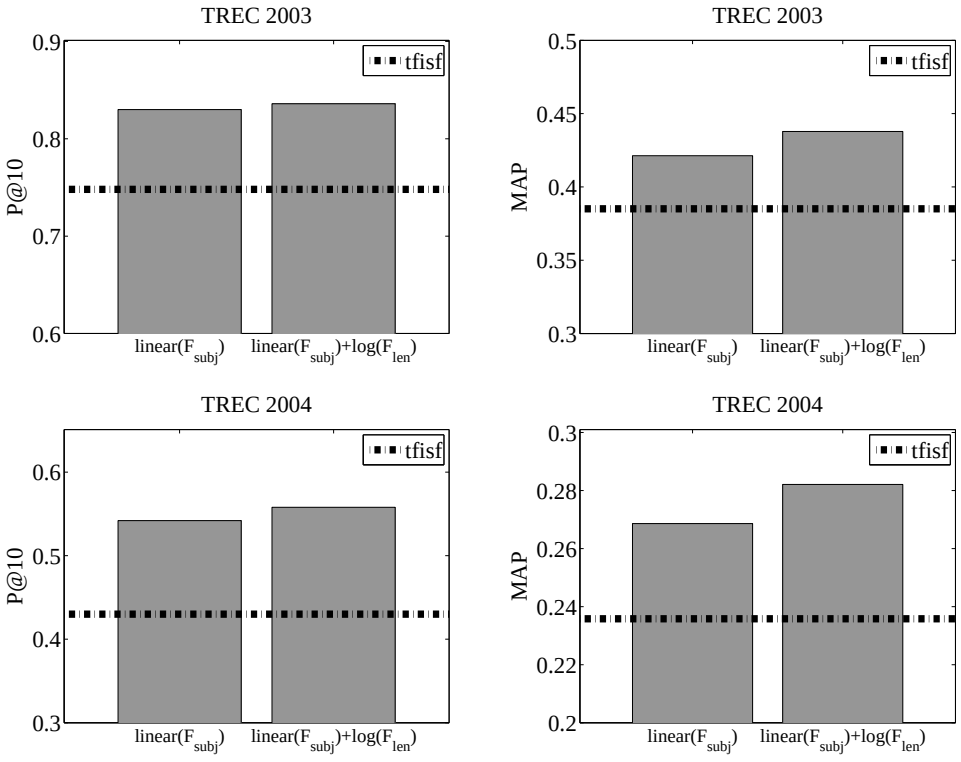


Figure 1.22: Comparison between the best opinion-based model (tfisf+linear(F_{subj})) and the best combination model (tfisf+linear(F_{subj})+log(F_{len})).

by including formally opinion-based weights (tfisf+linear(F_{subj})) and, additionally, the post-combination of sentence length weights on the top of the tfisf+linear(F_{subj}) model leads to further improvements in terms of MAP. Note that the improvements are very substantial (11-29%) and robust across collections.

1.5 Localized Smoothing and Sentence Importance

In this section we follow a different avenue to develop a more effective sentence retrieval method. We argue that the assumption engaged as a result of the naive application of document retrieval, i.e. that all sentences are independent, does not hold. This is because a sentence is surrounded by other sentences which help to contextualize it. Also the sentence is part of a document, and this sentence may or may not be important in representing the topic of the document. Presently, this *local context* is either ignored or underutilized by existing methods. We posit that, by incorporating the local context within SR models, more effective SR methods can be developed.

The reasons for this are as follows: any model using only standard term statistics to match query and sentences will suffer severely from the vocabulary mismatch problem because there is little overlap between the query and sentence terms. Intuitively, the local context could be used to improve retrieval, by helping to mitigate the difficulties posed by the vocabulary mismatch rooted in the sparsity of sentences. Additionally, current methods do not exploit the importance of a sentence in a document, which we posit is an important factor in determining the relevance of a sentence. A relevant sentence needs to be indicative of the query topic, but also representative and important in the context of the document, i.e. we assume that key statements within a document are more likely to be relevant.

To this aim, we propose a novel reformulation of the SR problem that includes the local context in a Language Modeling (LM) framework. Within this principled framework, it is possible to naturally include additional evidence into the smoothing process in order to enrich the representation of sentences. Also, the model provides a way to include a query-independent probability that encodes the importance of a sentence in a document. In a set of experiments performed over the TREC test collections, we compare the proposed models against existing SR models and demonstrate that using local context within a LM framework delivers retrieval performance that significantly outperforms the current state of the art in sentence retrieval.

1.5.1 Sentence Retrieval Models

In this section, we first outline the standard LM approach applied to the problem of SR. Then, we propose a novel reformulation which includes local context seamlessly and intuitively within the model.

1.5.1.1 SR with Language Models (Standard Method)

Language Models are probabilistic mechanisms to explain the generation of text [PC98]. The simplest LM is the unigram LM, which consists of associating a probability to each word of the vocabulary [ZL01, Hie01, MLS99]. This is a very intuitive and powerful approach that has been shown to be very effective in many IR tasks, such as ad-hoc retrieval [ZL01], distributed IR [SJCO02], and expert finding [BAdR09].

Given the SR problem, the idea is to estimate relevance according to the probability of generating a sentence s given the query q, expressed as $p(s|q)$. Instead of directly estimating this probability, Bayes Theorem is applied, and sentences can be ranked using the query-likelihood approach, $p(q|s)$[16]. The probability of a query q given the sentence s can then be estimated using the standard LM approach where, for each sentence s, a sentence LM is inferred. From the sentence model θ_s it is assumed that each query term t is sampled independently and identically, such that:

$$p(q|\theta_s) = \prod_{t \in q} p(t|\theta_s)^{c(t,q)} \tag{1.21}$$

where, $c(t, q)$ is the number of times the term t appears in q. The sentence model is constructed through a mixture between the probability of a term in the sentence and the probability of a term occurring in some background collection (i.e. maximum likelihood estimators of sentence and collection, respectively). This is usually performed in one of two ways by using a) Jelinek-Mercer (JM) smoothing as shown in Equation 1.22, or b) Dirichlet (DIR) smoothing as shown in Equation 1.23.

$$p(t|\theta_s) = (1 - \lambda) \cdot p(t|s) + \lambda \cdot p(t) \tag{1.22}$$

$$p(t|\theta_s) = \frac{c(t, s) + \mu \cdot p(t)}{c(s) + \mu} \tag{1.23}$$

where $c(t, s)$ is the number times that t appears in s, and $c(s)$ is the number of terms in the sentence. λ and μ are parameters that control the amount

[16]This assumes that there is not a priori preference for particular types of sentences, i.e. $p(s)$ is uniform.

of smoothing. Note that, in Equations 1.22 and 1.23, the smoothing expression ignores any local context and resorts immediately to the most general background knowledge $p(t)$. This is a strong assumption because it focuses the computation on sentence and collection statistics, without regard to any reference to other terms and phrases in sentences within the same document. As previously mentioned, many SR models [AWB03] take similar simplifications as the query-sentence similarity values do not take into account any information from the document, i.e. all sentences are treated independently.

JM and DIR smoothing yield to retrieval matching functions with specific length retrieval trends. In [LA08a] and [SA05], the authors studied these trends. In [LA08a], Losada and Azzopardi reported that DIR smoothing performs better than JM smoothing by showing that the document length pattern resembles the relevance pattern. They showed that DIR priors balance the query modeling and the document modeling roles, whereas JM smoothing does not consider the document length in the smoothing process. Thus, JM leads to poor retrieval performance because documents tend to be longer than the documents retrieved by DIR and the smoothing cannot compensate this. In [SA05], Smucker and Allan demonstrated that DIR smoothing's performance advantage arises from an implicit document prior that favors longer documents by smoothing them less. They tested the performance of a DIR prior and the JM smoothing with and without the document prior and showed that both methods smooth documents identically, except that the DIR prior smooths longer documents less. The result of this meant that the DIR prior tends to favor the retrieval of longer documents. Given the sentence retrieval problem, it is an open question as to what kind of length correction is appropriate for this task and whether the implicit length correction of smoothing methods employed help or hinder in the retrieval of relevant sentences.

1.5.1.2 Sentence Retrieval using Language Models with Local Context

In this section, we relax the independence assumption between sentences and assume that the document (i.e. the local context) plays an important role in determining the relevance of a sentence. Therefore, we treat the SR problem as a problem of estimating the probability of the query and the document given the sentence, i.e. is the sentence likely to be a generator of both the query and the document? This assumes that there is a correlation between this likelihood, $p(q, d|s)$ (where d is the document that contains s) and the relevance of the sentence. Thus, we posit that relevance is affected by how well the sentence explains both the document and the query topic

(as opposed to the query topic alone). In order to simplify the estimation of the conditional joint probability, we can rewrite it as follows:

$$p(q, d|s) = p(q|s, d) \cdot p(d|s) \tag{1.24}$$

where $p(q|s, d)$ is the probability of the query given the sentence and document, and $p(d|s)$ is the probability of the document given the sentence. Now we can clearly see that the estimation of the query likelihood will depend on both the sentence and the document. In addition, the $p(d|s)$ provides another way in which the local context is captured, by encoding the importance of a sentence within the document. In the next subsections we consider how these probabilities can be estimated.

1.5.1.3 Estimating $p(d|s)$

The probability of generating the document given the sentence, $p(d|s)$, can be regarded as a measure of the importance of the sentence within the topic of the document. Formally, this expression can be rewritten using Bayes' rule:

$$p(d|s) = \frac{p(s|d) \cdot p(d)}{p(s)} \tag{1.25}$$

where $p(s|d)$ is the probability of a sentence given a document, the $p(s)$ the probability of a sentence, and $p(d)$ is the prior probability of a document. Here, we assume that there is no a priori preference towards any of the documents, and treat $p(d)$ as a constant[17]. The probability $p(s|d)$ represents how likely the sentence is to be generated from the document, whereas $p(s)$ represents how likely the sentence is to be generated randomly. The ratio between the two expresses the importance of the sentence. Hence, in order to estimate $p(d|s)$, we compute $p(s)$ as:

$$p(s) = \prod_{t \in s} p(t)^{c(t,s)} \tag{1.26}$$

where $p(t)$ can be calculated using the maximum likelihood estimator of the term in a large collection: $p(t|\mathcal{C})$ (where $\mathcal{C}$ is the collection). Analogously, we define the probability of a sentence s given a document d as:

$$p(s|d) = \prod_{t \in s} p(t|d)^{c(t,s)} \tag{1.27}$$

[17]A simple alternative, which could be explored as part of future work, would be to estimate the prior based on the estimated relevance of the document.

where $p(t|d)$ is the probability of generating t from the maximum likelihood estimator of the document, and $c(t, s)$ is usually equals one as most terms only appear once in a sentence (unless the term is a stopword). It is to be noted that the problem of obtaining null probabilities from these estimates does not exist because terms that occur in a sentence will have non-zero probability in the LM of the document. Observe that $p(d|s)$ will give preference to those sentences that are central to the document's topics (i.e. high $p(s|d)$) but also rare within the collection (i.e. low $p(s)$). In this thesis we carefully study the effect of $p(d|s)$ on performance and have designed a complete set of experiments where we compare the estimation described above against the simplest (and naive) assumption: $p(d|s)$ is uniform.

1.5.1.4 Estimating $p(q|s, d)$

In order to estimate the query likelihood given the sentence and the document, we do this in a similar manner to the standard approach: first we assume that there is a model $\theta_{s,d}$ which generates the query terms, such that the probability of query given the sentence and the document is:

$$p(q|s, d) = \prod_{t \in q} p(t|\theta_{s,d})^{c(t,q)} \tag{1.28}$$

The LM $p(t|\theta_{s,d})$ is determined by the sentence and the local context denoted by d, thus we can represent the model as a mixture between the probability of a term in the sentence and the probability of a term in a document, which is then smoothed by the background model. The idea is that the terms in the document provide meaning to the sentence, and can improve the estimate of the relevance of a sentence.

For the time being, we assume that $p(t|d)$ is the normalized term frequency of t in d, but later we explore restricting this estimate to the sentences surrounding the sentence s.

There are several ways in which a mixture model can be defined using smoothing:

Three Mixture Model (3MM): The first model we propose here is a mixture of three LMs. This model assumes that queries are generated from a mixture of three different probability distributions: a LM for the sentence, $p(t|s)$, a LM for the document, $p(t|d)$, and a LM for the collection, $p(t|\mathcal{C})$ (or, simply, $p(t)$). Formally, we define this approach as:

$$p(t|\theta_{s,d}) = \lambda \cdot p(t|s) + \gamma \cdot p(t|d) + (1 - \lambda - \gamma) \cdot p(t) \tag{1.29}$$

where λ and γ are smoothing parameters such that λ,$\gamma \in [0,1]$. This estimator was initially proposed by Murdock in [Mur06]. Other authors have also applied 3MMs for other tasks, such as question-answering [XJC08]. Since the 3MM is very general, it is worth considering alternatives which smooth the sentence with the document and the collection but in a length-dependent way. This can be achieved by either first smoothing with the document proportionally to the sentence, and then interpolating with the collection (i.e. the Two Stage Model). Or, alternatively, first interpolating the sentence and the document, and then smoothing with the collection proportional to the sentence length. We shall detail these methods next.

Two-Stage Model (2S): The two-stage model adopted here is a variant of the well-known two-stage model used for document retrieval [ZL02]. This model is a combination of Dirichlet (DIR) and Jelinek-Mercer (JM) smoothing. Rather than smoothing with the collection model in both stages, we adapt here the model to the characteristics of the SR task and, therefore, the DIR stage uses $p(t|d)$ while the JM stage uses $p(t)$ for smoothing purposes. This is a simple and natural application of the two-stage smoothing for our problem. The formal expression is:

$$p(t|\theta_{s,d}) = (1-\lambda) \cdot \frac{c(t,s) + \mu \cdot p(t|d)}{c(s) + \mu} + \lambda \cdot p(t) \tag{1.30}$$

Two-Stage Model, Stages Inverted (2S-I): We propose here a two-stage model where the order in which DIR and JM smoothing methods are applied is inverted:

$$p(t|\theta_{s,d}) = \Big(1-\beta\Big) \cdot \Big((1-\lambda) \cdot p(t|s) + \lambda \cdot p(t|d)\Big) + \beta \cdot p(t) \tag{1.31}$$

where $\beta = \frac{\mu}{c(s)+\mu}$. The sentence model is first smoothed using linear interpolation with the document's model. Next, DIR is applied to smooth with the collection model[18]. By smoothing in this way, the first stage provides a new estimate of the foreground terms by combining the sentence and the document (through linear interpolation) and, then, the next stage adjusts the estimates with the background Language Model proportional to the length of the sentence. By inverting the smoothing methods, different length normalization schemes are applied to the sentence Language Models. In later sections, we shall analytically and empirically show how the 2S and 2S-I models differ in this respect.

[18]As shown in [ZL01], Dirichlet smoothing can be rewritten in a linear interpolation fashion with a proper document-dependent parameter.

Likelihood	Smoothing	Without $p(d\|s)$	With $p(d\|s)$
$p(q\|\theta_s)$	JM	[LC02, Los08, LF07]	untested
$p(q\|\theta_s)$	DIR	[LC02, Los08, LF07]	untested
$p(q\|\theta_{s,d})$	3MM	[Mur06]	untested
$p(q\|\theta_{s,d})$	2S	untested	untested
$p(q\|\theta_{s,d})$	2S-I	untested	untested

Table 1.12: Language Models included in our study. Most of the configurations are novel and have not been tested in the literature.

Observe that DIR and JM smoothing can also be included within this framework assuming that $p(q|s,d) = p(q|s)$ and applying DIR or JM to estimate the likelihood. If $p(d|s)$ is uniform, then these models are equivalent to the ones discussed in Section 1.5.1.1. However, if $p(d|s)$ is not uniform, then we get a novel combination of these popular smoothing strategies with the estimation of the importance of sentences in documents. Table 1.12 summarizes the different proposed models and informs about what configurations are novel (and, therefore, have not been tested in the literature).

In order to estimate relevance in this framework, we use tfisf and BM25 as sentence retrieval baselines. These models were introduced in Section 1.3. Although both models perform similarly. We include here BM25 as an additional baseline because, later on, we will use its field-based extension (BM25f) to incorporate the context.

1.5.2 Empirical Study

This section presents the experimental methodology employed to thoroughly evaluate the performance of the proposed models against existing and state of the art models. Particular attention is paid to examining the differences in performance brought about by the inclusion of the local context. Specifically, we hypothesize that:

1. localized smoothing will improve the estimate of the sentence models, resulting in improved effectiveness, and

2. the centrality of a sentence in a document helps to infer the relevance of a sentence, i.e. sentences that briefly summarize a document tend to be more relevant than the rest of sentences in the document.

1.5.2.1 Experimental Setup

For our experiments here, we report the performance of each method using precision at ten sentences (P@10) and Mean Average Precision (MAP). Observe that the models proposed are recall-oriented in nature, so we would expect to witness gains in terms of MAP. This is because the new models are able to promote sentences that do not necessarily match many query terms, but their context matches with some of the query terms. This should enhance the recall of relevant sentences (in particular sentences which may not overlap with the query terms). The usefulness of recall in sentence retrieval can be illustrated using the application scenario presented in the TREC Novelty Track [Har02], where a user is examining the ranked list of documents and is interested in reviewing *all* the on-topic sentences, but wants to skip through the non-relevant sentences. In this case, navigation could be made more efficient so that they can transverse through all the relevant sentences in all the documents. Whereas in the context of multi-document summarization, having access to all the relevant sentences is also very important. However, the precision-oriented measures (P@10) are also important for tasks likes query-biased summarization, snippet generation and question-answering. Ideally, the proposed models will be able to enhance both precision and recall based measures, but are likely to gain the largest improvements in terms of recall.

During the course of our experiments, each method presented in Section 1.5.1 was evaluated. Since many of the methods required parameter tuning, we ensured a fair comparison by employing a train-test methodology. Training of each method (except tfisf, which is parameter free) was performed on one of the three TREC datasets. For BM25 we considered the following range of values: k_1=1.0-2.0 (steps of 0.1), b=0.0-1.0 (steps of 0.1) and k_3 was fixed to 0 (the effect of k_3 is negligible with short queries). For the LM methods, λ was set to 0.1-0.9 (steps of 0.1), the range of values of μ (for 2S and 2S-I) was {1, 5, 10, 25, 50, 100, 250, 500, 1000, 2500, 5000, 10000} and the range of values for γ (for the 3MM model) was 0.1-0.9 (steps of 0.1). The parameter settings showing best performance were then fixed. These were then used to conduct the remainder of the evaluation, which was performed on the two remaining datasets. We experimented with the three possible training/testing configurations (training with TREC 2002 and testing with TREC 2003 and TREC 2004; training with TREC 2003 and testing with TREC 2002 and TREC 2004; and training with TREC 2004 and testing with TREC 2002 and TREC 2003) and found the same trends. In the next sections we report and discuss the results achieved by training with TREC 2002 and testing with TREC 2003 and TREC 2004. However, we include the results for the other training/testing configurations in Appendix B to further

demonstrate that our methods are robust.

Three models may be needed in order to estimate the relevance of a sentence: a sentence model, a local context model (where all the sentences in the document or the surrounding sentences where considered, depending on the type of the smoothing applied) and the background model (which is generated from all the documents in the collection).

When evaluating the LM approaches, we considered different alternatives. On one hand, we study the impact of $p(d|s)$ to specifically study the effect that this extra and novel component has on SR effectiveness. On the other hand, we considered two different contexts: the document (as it was shown in Section 1.5.1) and the surrounding sentences (see the below subsection).

Smoothing with Surrounding Sentences

In the previous sections we studied smoothing methods that included $p(t|d)$ within the sentence model, where $p(t|d)$ was estimated using the maximum likelihood estimate of a term in a document. This implies that all terms in the document are related to the sentence. Here, we propose an alternative estimate of $p(t|d)$ which relaxes this assumption and assumes that only the sentences surrounding the sentence being scored are related. So given a sentence s, the sentences immediately preceding and following s are directly related to it and, therefore, they constitute a closer context to the sentence s. In this way, considering the surrounding sentences only, a more accurate representation of the sentence LM should be obtained, which we anticipate will also lead to improved performance.

In this case, given a sentence s, its context c_s is composed by the previous sentence s_{prev}, the current sentence s and the next sentence in the document s_{next}[19]. Smoothing is performed by using $p(t|c_s)$ instead of $p(t|d)$ in Equations 1.29, 1.30 and 1.31, where $p(t|c_s)$ is the normalized count of t that occurs in s_{prev}, s and s_{next}.

In the next subsection we show the results of this approach and compare them against the results obtained when smoothing with documents instead of surrounding sentences.

1.5.2.2 Experimental Results

The first set of experiments tested the effect of localized smoothing *without* $p(d|s)$ (i.e. sentence importance is not considered, all sentences are considered as equally important). Then, we perform a second set of experiments that examines the impact of sentence importance. Finally, we present additional

[19]If s is the first or the last sentence in the document, then s_{prev} or s_{next} are ignored, respectively.

	P@10		**MAP**	
BM25	k_1=1.2, b=0, k_3=0		k_1=1.4, b=0, k_3=0	
	$\boldsymbol{p(q\|s,d)}$	$\boldsymbol{p(q\|s,c_s)}$	$\boldsymbol{p(q\|s,d)}$	$\boldsymbol{p(q\|s,c_s)}$
3MM	λ=0.1, γ=0.9	λ=0.7, γ=0.2	λ=0.8, γ=0.1	λ=0.8, γ=0.1
2S	λ=0.9, μ=250	λ=0.1, μ=500	λ=0.8, μ=5000	λ=0.1, μ=1
2S-I	λ=0.9, μ=10000	λ=0.8, μ=500	λ=0.9, μ=5000	λ=0.6, μ=500
DIR	μ=100		μ=500	
JM	λ=0.1		λ=0.1	

Table 1.13: Optimal parameter settings in the training collection (TREC 2002) for BM25 and LMs without p(d|s).

experiments to determine whether or not the baseline models can also be enhanced by including local context.

Influence of localized smoothing: Table 1.13 reports the parameter setting that optimized performance. Given the TREC 2002 as the training collection, Table 1.14 and Figure 1.23 show the performance in the test collections of the methods against the baselines in terms of P@10 and MAP. The Table shows the performance of models that use either the document as context, or the surrounding sentences. The best performance is presented in bold. Statistically significant differences between a given result and tfisf are marked with an asterisk, and statistically significant differences with respect to standard DIR smoothing are marked with a † (DIR provides the LM baseline, which is referred to as LMB). The test results obtained when TREC 2003 and TREC 2004 were used as the training collection are also provided in the Appendix B.

In Table 1.14 the first prominent result is that the 2S-I smoothing method is the best performing method in terms of MAP. This novel method is significantly better than the tfisf and DIR baselines, when either surrounding sentences or the entire document are used in the estimate. This is a good result, as it provides a simple and intuitive method that outperforms the long standing benchmark held on these standard test collections. The results in Tables B.2 and B.4, and Figures B.1 and B.2, also show similar improvements.

In terms of P@10, though, the performance of most of the contextually smoothed models is slightly poorer than the baselines. The 2S-I method does provide the best performance at P@10 on the TREC 2004 collection when using the surrounding sentences to smooth the Language Models. However, though this is not always significantly different from the baselines.

As previously mentioned, this is perhaps to be expected because the pro-

Context					Document			Surrounding Sents.		
			$p(q\|s)$		$p(q\|s,d)$			$p(q\|s,c_S)$		
	tfisf	**BM25**	**DIR (LMB)**	**JM**	**3MM**	**2S**	**2S-I**	**3MM**	**2S**	**2S-I**
					TREC 2003					
P@10	.7480	**.7540**†	.6960*	.5600*†	.5020*†	.5680*†	.7080	.5200*†	.4480*†	.7320
Δ% (tfisf)		*(+0.80)*	*(-6.95)*	*(-25.13)*	*(-32.89)*	*(-24.06)*	*(-5.35)*	*(-30.48)*	*(-40.11)*	*(-2.14)*
Δ% (LMB)	*(+7.47)*	*(+8.33)*		*(-19.54)*	*(-27.87)*	*(-18.39)*	*(+1.72)*	*(-25.29)*	*(-35.63)*	*(+5.17)*
MAP	.3851†	.3852†	.3638*	.3474*†	.3513*†	.3502*	**.4099***†	.3532*†	.3494*†	.3893†
Δ% (tfisf)		*(+0.03)*	*(-5.53)*	*(-9.79)*	*(-8.78)*	*(-9.06)*	*(+6.44)*	*(-8.28)*	*(-9.27)*	*(+1.09)*
Δ% (LMB)	*(+5.85)*	*(+5.88)*		*(-4.51)*	*(-3.44)*	*(-3.74)*	*(+12.67)*	*(-2.91)*	*(-3.96)*	*(+7.01)*
					TREC 2004					
P@10	.4300	.4380	.4200	.3580*†	.2940*†	.3540*†	.4300	.3420*†	.2720*†	**.4700***
Δ% (tfisf)		*(+1.86)*	*(-2.33)*	*(-16.74)*	*(-31.63)*	*(-17.67)*	*(+0.00)*	*(-20.47)*	*(-36.74)*	*(+9.30)*
Δ% (LMB)	*(+2.38)*	*(+4.29)*		*(-14.76)*	*(-30.00)*	*(-15.71)*	*(+2.38)*	*(-18.57)*	*(-35.24)*	*(+11.90)*
MAP	.2358†	.2368*†	.2240*	.2131*†	.2195*	.2203*	**.2550***†	.2226*	.2204*	.2488*†
Δ% (tfisf)		*(+0.42)*	*(-5.00)*	*(-9.63)*	*(-6.91)*	*(-6.57)*	*(+8.14)*	*(-5.60)*	*(-6.53)*	*(+5.51)*
Δ% (LMB)	*(+5.27)*	*(+5.71)*		*(-4.87)*	*(-2.01)*	*(-1.65)*	*(+13.84)*	*(-0.63)*	*(-1.61)*	*(+11.07)*

Table 1.14: P@10 and MAP in the test collections (TREC 2003 & TREC 2004) without sentence importance. Statistically significant differences with respect to tfisf are marked with * and with respect to LMB are marked with †.

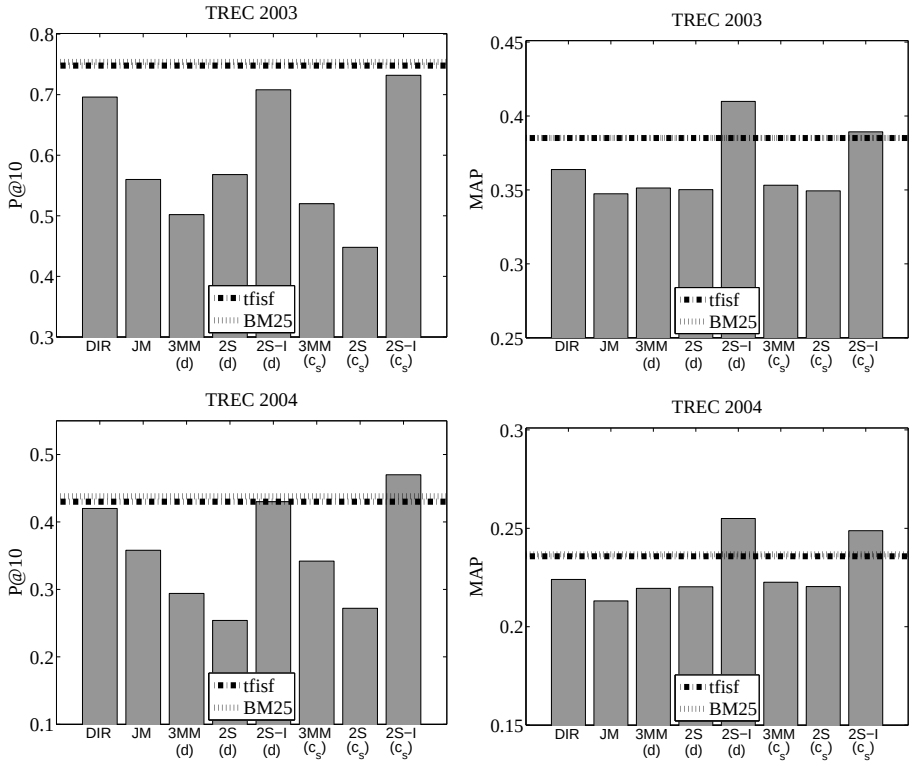


Figure 1.23: P@10 and MAP in the test collections (TREC 2003 & TREC 2004) without sentence importance.

	P@10		**MAP**	
BM25	k_1=1.2, b=0, k_3=0		k_1=1.4, b=0, k_3=0	
	$p(q\|s,d)p(d\|s)$	$p(q\|s,c_s)p(d\|s)$	$p(q\|s,d)p(d\|s)$	$p(q\|s,c_s)p(d\|s)$
3MM	λ=0.3, γ=0.3	λ=0.1, γ=0.1	λ=0.5, γ=0.1	λ=0.6, γ=0.3
2S	λ=0.1, μ=1	λ=0.2, μ=1000	λ=0.1, μ=1	λ=0.1, μ=5
2S-I	λ=0.1, μ=1	λ=0.2, μ=250	λ=0.1, μ=10	λ=0.4, μ=1
DIR	μ=250		μ=1	
JM	λ=0.9		λ=0.1	

Table 1.15: Optimal parameter settings in the training collection (TREC 2002) for LMs with p(d|s).

posed methods are more likely to improve recall. Still, it is very encouraging to see that early precision can also be increased if the smoothing parameters are appropriately set. Recall that we have trained the parameters on a held out test collection, so the performance reported here is not necessarily the best that could be obtained using improved parameter estimation methods. For the remaining of this section, the focus of the discussion will be on performance with respect to MAP, unless otherwise specified.

In terms of the type of smoothing, i.e. using surrounding sentences or documents, there was no significant differences between the performance obtained with the different estimates. Though, using the complete document was slightly better overall. The other notable point is that the 3MM and 2S localized smoothing methods did not provide improvements to performance. This suggests that the 2S-I smoothing method provides an advantage over these other smoothing methods, which may not necessarily be because of the local information used. We explore the reasons in subsection 1.5.2.3.

Impact of Sentence Importance: In this set of experiments we considered the influence of the local context stemming from the importance of a sentence within a document. Table 1.15 reports the best settings in the training collections for the proposed LM methods with the sentence importance component. The performance of each method is shown in Table 1.16 and Figure 1.24 while Figure 1.25 provides a comparative bar graph of the P@10 and MAP of each method with and without $p(d|s)$. It is clear from these results that the inclusion of the sentence importance results in significantly better retrieval performance for all the LMs over the state of the art method (tfisf). It appears that the impact of the sentence importance dominates the localized smoothing. For instance, given the query "*Chinese earthquake*", the 3MM with sentence importance is able to retrieve the following relevant sentence within the top-10 sentences: "*Chinese architects from*

Context	Sentence Only				Document			Surrounding Sents.		
			$p(q\|s)p(d\|s)$		$p(q\|s,d)p(d\|s)$			$p(q\|s,c_s)p(d\|s)$		
	tfisf	**BM25**	**DIR**	**JM**	**3MM**	**2S**	**2S-I**	**3MM**	**2S**	**2S-I**
					TREC 2003					
P@10	.7480†	**.7540**†	.7280	.7320	.7220	.7440	.7360	.7260	.7340	.7280
Δ% (tfisf)		*(+0.8)*	*(-2.67)*	*(-2.14)*	*(-3.48)*	*(-0.53)*	*(-1.60)*	*(-2.94)*	*(-1.87)*	*(-2.67)*
Δ% (LMB)	*(+7.47)*	*(+8.33)*	*(+4.60)*	*(+5.17)*	*(+3.74)*	*(+6.90)*	*(+5.75)*	*(+4.31)*	*(+5.46)*	*(+4.60)*
MAP	.3851†	.3852†	**.4144***†	.4137*†	.4104*†	.4117*†	.4108*†	.4129*†	.4132*†	.4132*†
Δ% (tfisf)		*(+0.03)*	*(+7.61)*	*(+7.43)*	*(+6.57)*	*(+6.91)*	*(+6.67)*	*(+7.22)*	*(+7.30)*	*(+7.30)*
Δ% (LMB)	*(+5.85)*	*(+5.88)*	*(+13.91)*	*(+13.72)*	*(+12.81)*	*(+13.17)*	*(+12.92)*	*(+13.50)*	*(+13.58)*	*(+13.58)*
					TREC 2004					
P@10	.4300	.4380	.4380	**.4420**	.4400	**.4420**	.4380	.4400	.4380	.4380
Δ% (tfisf)		*(+1.86)*	*(+1.86)*	*(+2.79)*	*(+2.33)*	*(+2.79)*	*(+1.86)*	*(+2.33)*	*(+1.86)*	*(+1.86)*
Δ% (LMB)	*(+2.38)*	*(+4.29)*	*(+4.29)*	*(+5.24)*	*(+4.76)*	*(+5.24)*	*(+4.29)*	*(+4.76)*	*(+4.29)*	*(+4.29)*
MAP	.2358†	.2368*†	.2549*†	.2548*†	.2527*†	.2538*†	.2529*†	.2550*†	.2550*†	**.2553***†
Δ% (tfisf)		*(+0.42)*	*(+8.10)*	*(+8.06)*	*(+7.17)*	*(+7.63)*	*(+7.25)*	*(+8.14)*	*(+8.14)*	*(+8.27)*
Δ% (LMB)	*(+5.27)*	*(+5.71)*	*(+13.79)*	*(+13.75)*	*(+12.81)*	*(+13.30)*	*(+12.90)*	*(+13.84)*	*(+13.84)*	*(+13.97)*

Table 1.16: P@10 and MAP in the test collections (TREC 2003 & TREC 2004) after incorporating sentence importance ($p(d|s)$). Statistically significant differences with respect to tfisf are marked with * and with respect to standard DIR (LMB) are marked with †.

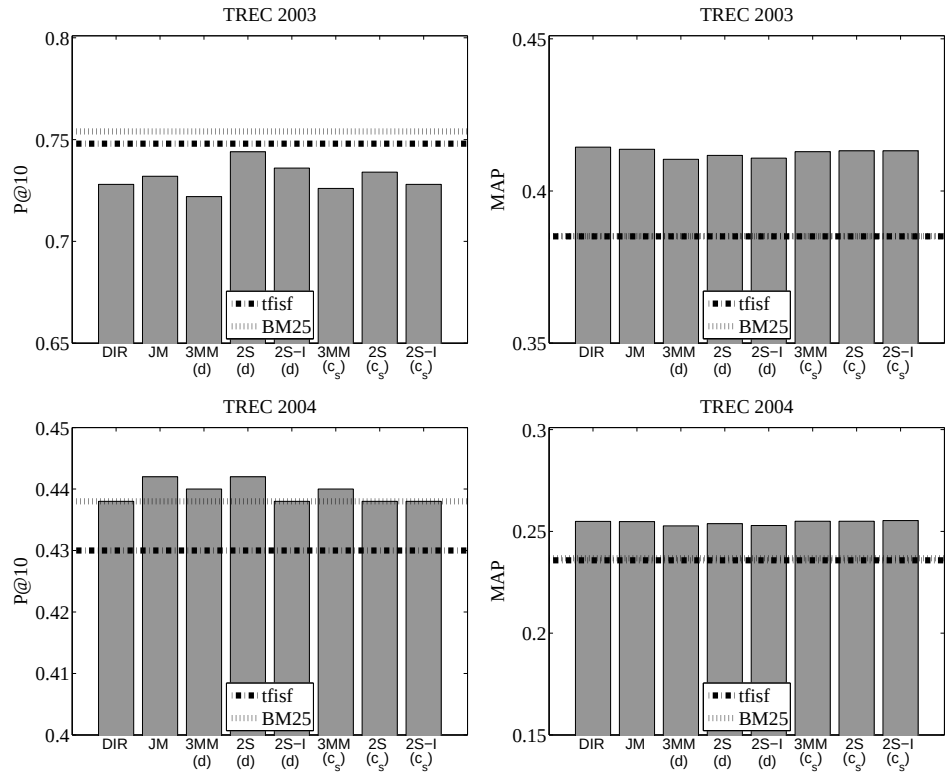


Figure 1.24: P@10 and MAP in the test collections (TREC 2003 & TREC 2004) with $p(d|s)$.

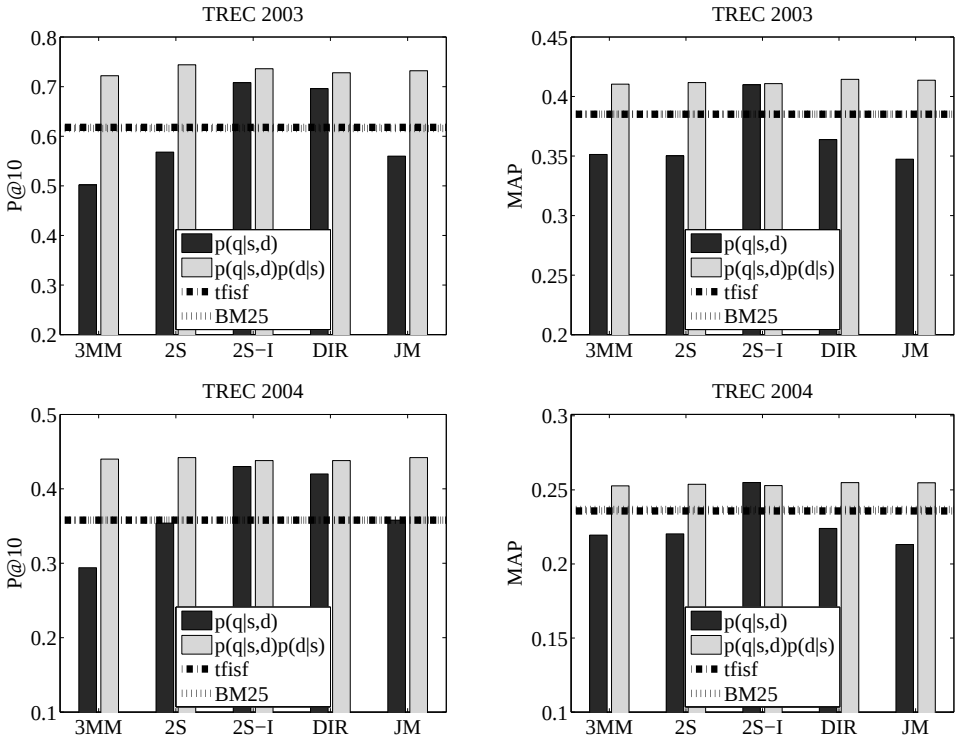


Figure 1.25: P@10 and MAP in the test collections (TREC 2003 & TREC 2004) of the LMs with and without sentence importance.

the Ministry of Construction and Hebei Province and the city of Zhangjiakou have begun work on rebuilding earthquake-damaged parts of Hebei and have completed design work on ten types of residential housing for nine villages as models". Nevertheless, this sentence does not appear in the top-10 of the version of 3MM that does not include sentence importance. This is because this sentence summarizes well the document and, therefore, the $p(d|s)$ factor promotes it.

There are not significantly different levels of effectiveness between each of the different smoothing methods. Observe also that the performance of 2S-I is not substantially affected by the sentence importance factor.

All the models that include $p(d|s)$ are novel, as previous proposals using LMs are solely based on query likelihood estimations. Note also that the three-mixture model as proposed in [Mur06] (i.e. without $p(d|s)$) performs worse than the strong and weak baselines (results shown in the 5^{th} column of Table 1.14).

Incorporating Context into the Baselines: The baseline models (tf-isf and BM25) are context-unaware with respect to the local context. Given

the findings we have obtained from incorporating local context in the LM framework, it is natural to wonder whether introducing the local context into the baselines can also improve their performance. First, we present several straightforward adaptations of BM25 and tfisf to include local context, then we compare these variations under the same experimental conditions as above.

A natural solution to introduce document statistics into BM25 is to use the extended version of this model to handle multiple weighted fields, i.e. BM25f [RZT04]. BM25f estimates the relevance of documents considering a document as a set of components. Each of these components may be assigned a specific weight within the document. For our case, a sentence (s) can be considered as an aggregate of the sentence itself and the context containing the sentence (i.e. the document or the surrounding sentences provide local context to the sentence). Given these two components, the BM25f model can be instantiated as follows:

$$sim_{\text{BM25f}}(s,q) = \sum_{t \in q \cap s} \log \frac{N - sf(t) + 0.5}{sf(t) + 0.5} \cdot \frac{weight(t,s)}{k_1 + weight(t,s)} \cdot \frac{(k_3+1)\cdot c(t,q)}{k_3 + c(t,q)} \tag{1.32}$$

$$weight(t,s) = \frac{c(t,s)\cdot\alpha}{(1-b_{sen}) + b_{sen}\cdot\frac{c(s)}{avsl}} + \frac{c(t,context)\cdot(1-\alpha)}{(1-b_{context}) + b_{context}\cdot\frac{c(context)}{avcl}} \tag{1.33}$$

where b_{sent} and $b_{context}$ are normalizing constants associated to the field length in s and its context, respectively; α is a boost factor that controls the term frequency mixture between context statistics and sentence statistics; $c(context)$ ($c(s)$) is the number of terms in context (s), $c(t, context)$ is either $c(t,d)$ or $c(t,c_s)$ (depending on whether we apply document-level or surrounding sentences context), and $avcl$ ($avsl$) is the average context (sentence) length in the collection. To reduce the number of parameters to be tuned, $b_{context}$ was fixed to 0.75 (the value usually recommended for document length normalization in BM25 [Rob05]), k_1 was set to the optimal value found with BM25 (Table 1.13) and k_3 was set again to 0. The remaining parameters, α and b_{sen}, were tuned in the training collection (ranging from 0 to 1 in steps of 0.1).

Regarding tfisf, no extensions have been defined to handle local context and, therefore, we defined ad-hoc adjustments to mix context statistics with sentence statistics. We tested the following variants of tfisf:

	BM25	**BM25f**	
		BM25f(d)	**BM25f(c_s)**
		$b_{sen}=0$, $\alpha=1$	$b_{sen}=0$, $\alpha=1$
		TREC 2003	
P@10	.7540	.7540	.7540
Δ%		*(+0.0)*	*(+0.0)*
MAP	.3852	.3852	.3852
Δ%		*(+0.0)*	*(+0.0)*
		TREC 2004	
P@10	.4380	.4380	.4380
Δ%		*(+0.0)*	*(+0.0)*
MAP	.2368	.2368	.2368
Δ%		*(+0.0)*	*(+0.0)*

Table 1.17: Performance of BM25 and its variations (BM25f) to include context in the test collections (TREC 2003 & TREC 2004).

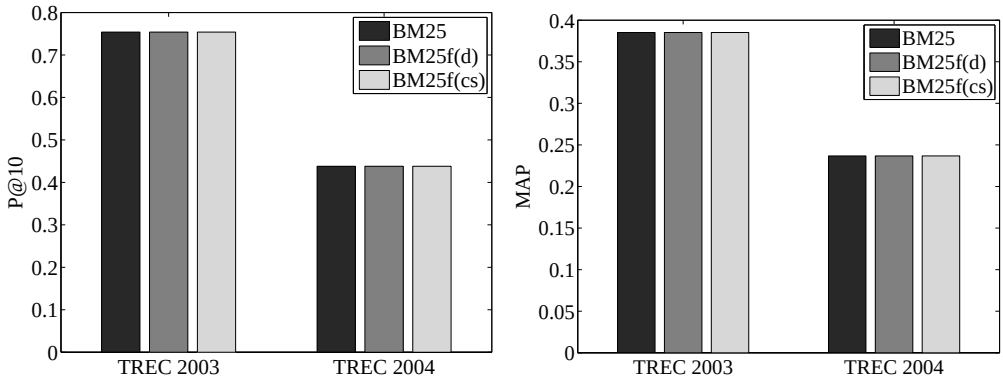


Figure 1.26: Performance of BM25 and its variations (BM25f) to include context in the test collections (TREC 2003 & TREC 2004).

a) tfmix: c(t,s) is replaced by $\alpha \cdot c(t,s) + (1-\alpha) \cdot c(t, context)$;

b) idfdoc: $sf(t)$ is replaced by $df(t)$ (i.e. idf is computed at the document level rather than at sentence level);

c) tfmix+idfdoc: where both a) and b) were applied.

At training time, only α needs to be tuned (between 0 and 1 in steps of 0.1). Again, TREC 2002 was the training collection and TREC 2003 and TREC 2004 were the test collections. The optimal performance was reached with $b_{sen} = 0$ and $\alpha = 1$ (BM25f), and $\alpha = 1$ (tfisf). This means that these models obtain best performance, when the local context is largely ignored! Tables 1.17 and 1.18 and Figures 1.26 and 1.27 report the results achieved

			tfmix		tfmix+idfdoc	
	tfisf	**idfdoc**	**tfmix(d)** $\alpha = 1$	**tfmix(c_s)** $\alpha = 0.6$	**tfmix+idfdoc(d)** $\alpha = 1$	**tfmix+idfdoc(c_s)** $\alpha = 0.6$
			TREC 2003			
P@10	.7480	.7540	.7480	.7380	.7540	.7480
Δ%		*(+0.80)*	*(+0.00)*	*(-1.34)*	*(+0.80)*	*(+0.00)*
MAP	.3851	.3906*	.3851	.3843	.3906	.3843
Δ%		*(+1.43)*	*(+0.00)*	*(-0.21)*	*(+1.43)*	*(-0.21)*
			TREC 2004			
P@10	.4300	.4360	.4300	.4240	.4360	.4360
Δ%		*(+1.40)*	*(+0.00)*	*(-1.40)*	*(+1.40)*	*(+1.40)*
MAP	.2358	.2363	.2358	.2359	.2363	.2375
Δ%		*(+0.21)*	*(+0.00)*	*(+0.04)*	*(+0.21)*	*(+0.72)*

Table 1.18: Performance of tfisf and its variations to include context in the test collections (TREC 2003 & TREC 2004).

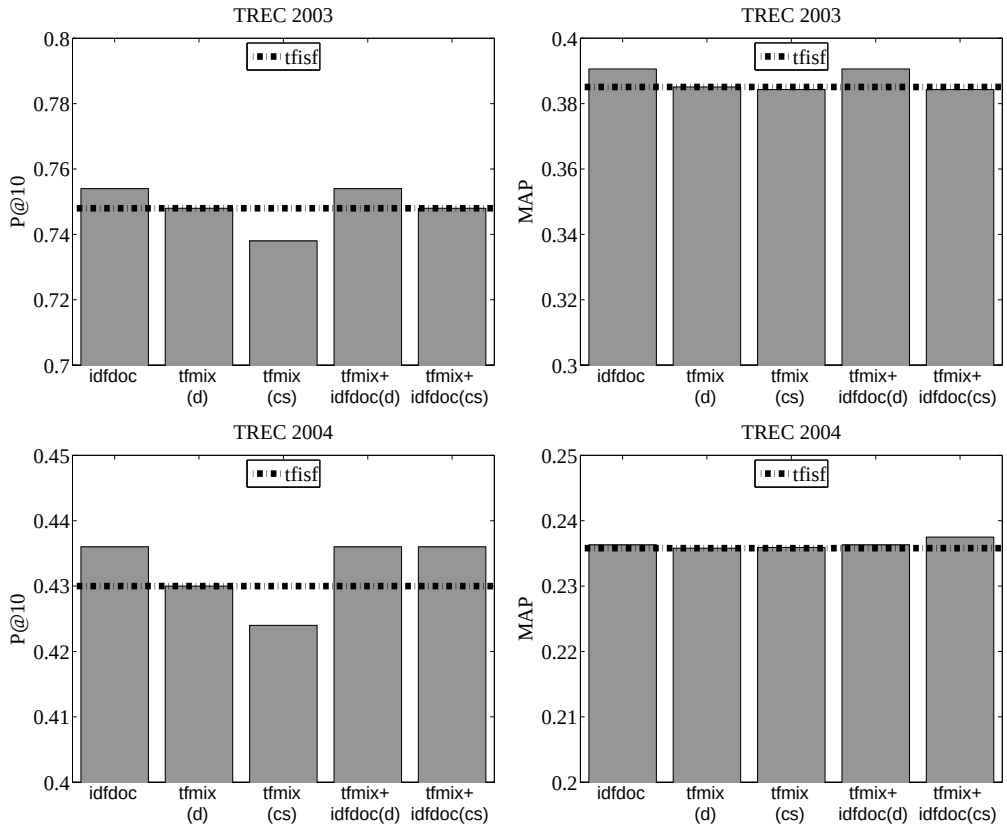


Figure 1.27: Performance of tfisf and its variations to include context in the test collections (TREC 2003 & TREC 2004).

in the test collections. Not surprisingly, the variations perform virtually the same as the original models. As a matter of fact, BM25f with $\alpha = 1$ (considering either the surrounding sentences or the document as a local context)

yields the same SR strategy as BM25. The same happens for tfisf+tfmix ($\alpha = 1$) with respect to tfisf when the document is considered as the local context. Nevertheless, tfisf+tfmix considering the surrounding sentences ($\alpha = 0.6$) performs worse than tfisf in TREC 2003 and the same as tfisf in TREC 2004. With idfdoc there are some slight variations in performance with respect to the baseline but they are insignificant[20].

While it appears that local context can be useful, the model in which it is incorporated determines how successfully this evidence can be used. In the Language Modeling approach, the framework provides a natural and intuitive manner to encode and incorporate the local context through the smoothing process. However, it is unclear how to effectively incorporate the evidence within these other models. We leave this direction for future work, and study more precisely why and how the Language Models are able to capitalize on this additional evidence.

1.5.2.3 Analysis

In this section, we conduct a detailed analysis to understand precisely the reasons behind the differences in effectiveness of the LMs designed. To explain the improvements in performance brought about by the 2S-I model when no sentence importance is used, we derived the retrieval formulas associated to these LMs (similar to that performed in [ZL01, LA08b]). The retrieval formulas in sum-log form are shown in Table 1.19. Examining the models in this way we can see the differences between each smoothing method. It is interesting to pay attention to the second addend in these formulas. This component incorporates usually some form of length correction. In the DIR and 2S method, this component penalizes long sentences and acts as a length normalization component (which is useful for document retrieval)[21] [LA08a]. In the JM and 3MM methods, this component is independent to the length of the sentence. However, in the 2S-I method, this component *promotes long sentences* because a high $c(s)$ means that β is low making that, overall, the sum is greater (because, usually, $p(t|d) >> p(t)$).

To illustrate this point further, in Figure 1.28 we show the behavior of the length correction that the DIR, 2S and 2S-I methods produce with respect to the sentence length. Such correction is given by the second addend

[20] We also tried other values of α on the test collections and can confirm that when $\alpha = 1$ and $\alpha = 0.6$ the best performance was obtained when the document or the surrounding sentences are considered, respectively.

[21] Note that since older retrieval models such as tf and tf-idf using a vector space model overly favored longer documents, a length correction was required, which penalized longer documents. However, in sentence retrieval it would appear this is not appropriate.

Model	Retrieval formula
DIR	$\sum_{t \in s \cap q} c(t,q) \cdot \log\left(1 + \frac{c(t,s)}{\mu \cdot p(t)}\right) + c(q) \cdot \log \frac{\mu}{c(s)+\mu}$
JM	$\sum_{t \in s \cap q} c(t,q) \cdot \log\left(1 + \frac{(1-\lambda)}{\lambda} \cdot \frac{c(t,s)}{c(s) \cdot p(t)}\right) + c(q) \cdot \log \lambda$
3MM	$\sum_{t \in s \cap q} c(t,q) \cdot \log \frac{\lambda \cdot p(t\|s) + \gamma \cdot p(t\|d) + (1-\lambda-\gamma) \cdot p(t)}{\gamma \cdot p(t\|d) + (1-\lambda-\gamma) \cdot p(t)}$ $+ \sum_{t \in q} c(t,q) \cdot \log(\gamma \cdot p(t\|d) + (1-\lambda-\gamma) \cdot p(t))$
2S	$\sum_{t \in s \cap q} c(t,q) \cdot \log \frac{(1-\lambda) \cdot \frac{c(t,s)+\mu \cdot p(t\|d)}{c(s)+\mu} + \lambda p(t)}{(1-\lambda)\frac{\mu \cdot p(t\|d)}{c(s)+\mu} + \lambda \cdot p(t)}$ $+ \sum_{t \in q} c(t,q) \cdot \log((1-\lambda) \cdot \frac{\mu \cdot p(t\|d)}{c(s)+\mu} + \lambda \cdot p(t))$
2S-I	$\sum_{t \in s \cap q} c(t,q) \cdot \log \frac{(1-\beta) \cdot ((1-\lambda) \cdot p(t\|s) + \lambda \cdot p(t\|d)) + \beta \cdot p(t)}{(1-\beta) \cdot \lambda \cdot p(t\|d) + \beta \cdot p(t)}$ $+ \sum_{t \in q} c(t,q) \cdot \log((1-\beta) \cdot \lambda \cdot p(t\|d) + \beta \cdot p(t))$ $(\beta = \mu/(c(s)+\mu))$

Table 1.19: Sum-log retrieval formulas for the SR models based on LMs (without $p(d|s)$).

of expressions in Table 1.19. In this example, a query q with three terms (q_A, q_B, q_C) is used, where $c(q_A, q) = c(q_B, q) = c(q_C, q) = 1$, $p(q_A) = 10^{-6}$, $p(q_B) = 10^{-12}$, $p(q_C) = 10^{-3}$, $p(q_A|d) = p(q_B|d) = p(q_C|d) = 10^{-2}$, $\lambda = 0.5$, $\mu = 100$. Then the sentence length was varied from 1 to 50 (in steps of 1). Note that in DIR and 2S the correction factor decreases with sentence length, while in 2S-I the value of this factor increases with sentence length. This illustrates graphically that DIR and 2S methods are likely to promote short sentences, while the 2S-I method is likely to promote long sentences.

This seems to indicate that promoting long sentences is a way to achieve better performance, as opposed to using more information. Observe also that the best parameter setting in BM25 fixes b to 0 (Table 1.13), meaning

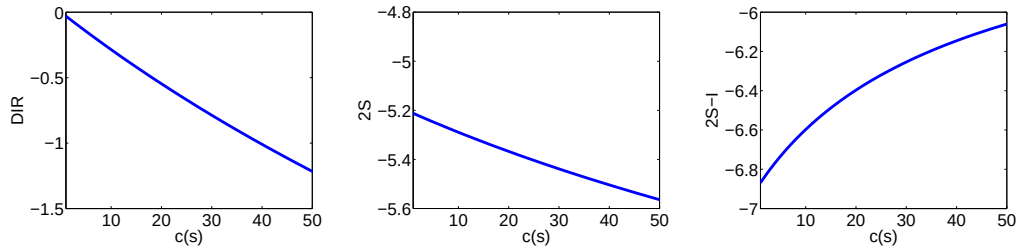


Figure 1.28: Effect of non-matching component (length correction) in DIR, 2S and 2S-I against sentence length. The plots show that the score assigned to sentences are adjusted proportionally to the length of the sentence. Note that the 2S-I method favors longer sentences, while the other methods penalize longer sentences.

that sentences are not penalized because of their length. To further support this claim, we analyzed the average length of sentences in these collections and compared it to the average length of relevant sentences. The average sentence length is around 9 terms in all collections, while the average length of relevant sentences is around 14 terms. Furthermore, we analyzed the top 100 sentences retrieved by every model and found that 2S-I yields an average length of 13.71 and 13.66 (TREC 2003 and TREC 2004, respectively), while the other models retrieve shorter sentences on average (e.g. 3MM retrieves sentences whose average length is 12.68 and 12.67, respectively). These statistics suggest that 2S-I is superior to the other models because it promotes longer sentences, and this is required to achieve better performance for the task of sentence retrieval.

Further to this analysis, it is interesting to note that in the estimation of $p(d|s)$ longer sentences will also attract a higher probability. As a matter of fact, in Table 1.20 and Figure 1.29 we compare the performance of DIR and JM methods and a variant of them consisting of incorporating a sentence length prior. We show that this variant outperforms significantly their corresponding original versions. However, as we show in Figure 1.30, these approaches do not outperform the 2S-I model (in terms of MAP) and, therefore, the sentence length is not the only component that makes the 2S-I model effective.

Observe that $p(d|s)$, as estimated in Section 1.5.1.3, is a factor that favors long sentences (because, for the vast majority of the terms in a sentence, $p(t|d) >> p(t)$[22]). This explains why 2S-I does not receive any significant benefits from $p(d|s)$ (as 2S-I already retrieves many long sentences) while the other LM techniques receive significant increases. As a matter of fact,

[22]Recall that $p(\cdot|d)$ and $p(\cdot)$ are both maximum likelihood estimators.

	$p(q\|s)$		$p(q\|s)p(s)$	
	DIR	**JM**	**DIR+len**	**JM+len**
	TREC 2003			
P@10	.6960	.5600	**.7500***	.6120*
	($\mu = 100$)	($\lambda = 0.1$)	($\mu = 250$)	($\lambda = 0.8$)
MAP	.3638	.3474	**3998***	3730*
	($\mu = 500$)	($\lambda = 0.1$)	($\mu = 50$)	($\lambda = 0.3$)
	TREC 2004			
P@10	.4200	.3580	**.4960***	.3740
	($\mu = 100$)	($\lambda = 0.1$)	($\mu = 250$)	($\lambda = 0.8$)
MAP	.2240	.2131	**2517***	2298*
	($\mu = 500$)	($\lambda = 0.1$)	($\mu = 50$)	($\lambda = 0.3$)

Table 1.20: Comparative between DIR and JM against their variants with the sentence length prior (trained with TREC 2002 and tested with TREC 2003 and TREC 2004).

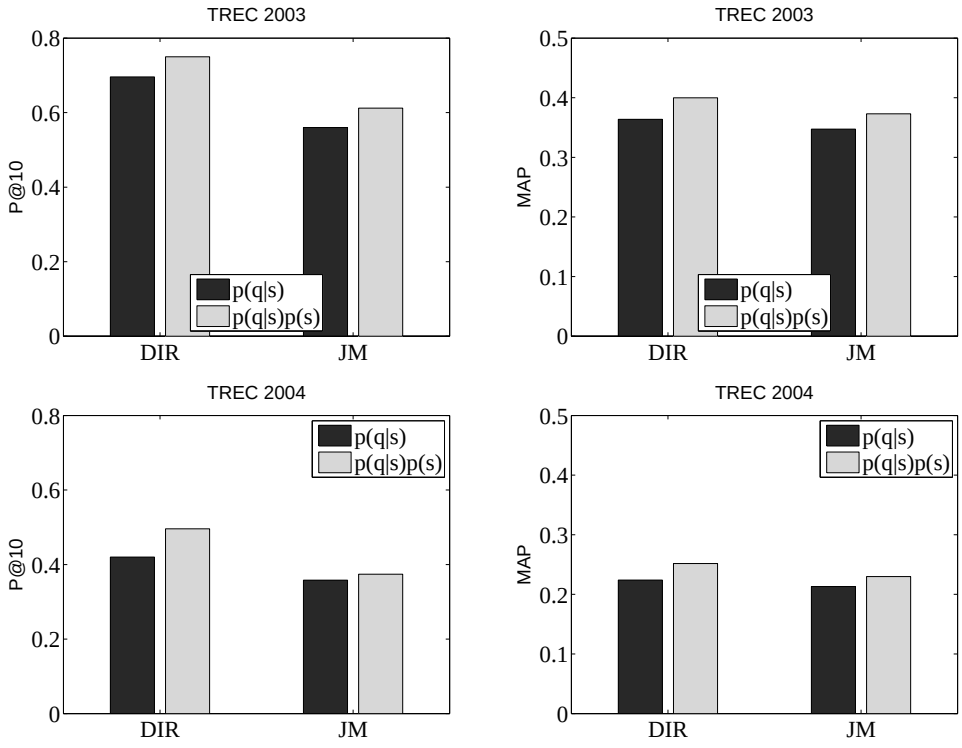


Figure 1.29: Comparative between DIR and JM against their variants considering a sentence length prior (trained with TREC 2002 and tested with TREC 2003 and TREC 2004).

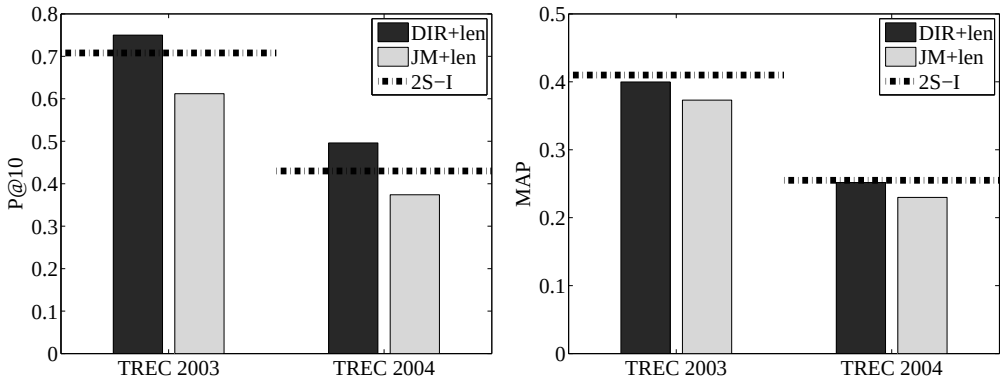


Figure 1.30: Comparison between LMs+length and 2S-I model (TREC 2003 & TREC 2004).

analyzing the top 100 sentences retrieved by every method with $p(d|s)$, we found that the average lengths are quite uniform across models (around 20 terms). This analysis suggests that the local context used indirectly promotes longer sentences, which results in improved retrieval effectiveness.

Summary and Discussion: To sum up, the importance of sentences within documents, $p(d|s)$, makes that the performance of the LMs improve significantly beyond existing state of the art methods. When ignoring $p(d|s)$, 2S-I is the only approach that handles well the retrieval of long sentences with document-level smoothing.

It is quite remarkable that any LM method with $p(d|s)$ is superior to the baselines. This suggests that retrieval methods such as tfisf and BM25 are limited because they are simple adaptations of document retrieval techniques and, therefore, they involve some sort of correction to avoid retrieving many long texts (e.g. b in BM25) but they do not have the opposite tool: some correction to retrieve more long texts. Standard models without length normalization (tfisf or BM25 setting b to 0) have already some tendency towards long pieces of text (because long sentences match more terms) but, given our findings, this is not sufficient to improve the model's performance. However, this also opens the door to future developments, or extensions of current SR models, to try to account for this tendency. This will also help to further understand the reason behind such good performance. As a matter of fact, the positive effect on tfisf of the query-independent weights based on sentence length proposed in Section 1.4.2 agrees with this line of thought.

1.6 Combining Query-Independent Features and Local Context

In previous sections we studied the impact of query-independent features and local context independently. We found that these two paths lead to models able to outperform the state of the art in sentence retrieval. In this section, we combine these strategies and study whether or not the combination is beneficial with respect to the models working in isolation. Observe that query-independent information, as introduced in Section 1.4, and local context (Section 1.5) are arguably two complementary ways to improve sentence retrieval performance. We focus here on the most effective query-independent features (opinion-based) in combination with the best contextual models.

In order to combine query-independent features with the local context, we consider the model that performs the best, i.e. the 2S model with $p(d|s)$. Named entity-based features are not considered here because they lead to insignificant improvements in performance, and sentence length is also skipped here because, as demonstrated in Section 1.5.2.3, the 2S model with $p(d|s)$ inherently incorporates sentence length, i.e. the estimation of relevance for a sentence is directly influenced by its length. To combine local context with opinion-based features we simply apply FLOE on the top of a baseline that incorporates context.

In Figure 1.31 we show the curves of the probabilities $p(I)$, $p(I|R)$ and $p(I|T)$, where T is the set of sentences retrieved by the 2S model (with $p(d|s)$) and I is one of these features: F_{subj}, F_{neg}, F_{pos}, F_{opt}. Given the F_{subj} feature, $p(I = 0|T) > p(I = 0|R)$ and $p(I = 1|R) < p(I = 1|R)$, meaning that our SR model is not retrieving enough subjective relevant sentences. For the remaining features there are not large distinctions between $p(I|R)$ and $p(I|T)$. Therefore, we anticipate that improvements after incorporating these features into the original SR model might not be attainable. Observe that this initial analysis of the context-based baseline reveals trends that are similar to those found with tfisf: the retrieval model does not retrieve enough subjective sentences. Still, the deviations shown here look smaller (e.g. compare the F_{subj} plot in Figure 1.31 with the F_{subj} plot in Figure 1.5) and, therefore, there might be less room for improvement.

In Figure 1.32 we show the adjustment under the independence assumption given different opinion-based features and, in Figure 1.33, we plot the adjustment proposed by FLOE against indep and $\log \frac{p(I|T)}{p(I)}$. We report in Table 1.21 and Figure 1.34 the performance of the 2S with $p(d|s)$ model after incorporating the different opinion-based features. The incorporation of opinion-based features into the context-based model leads to insignifi-

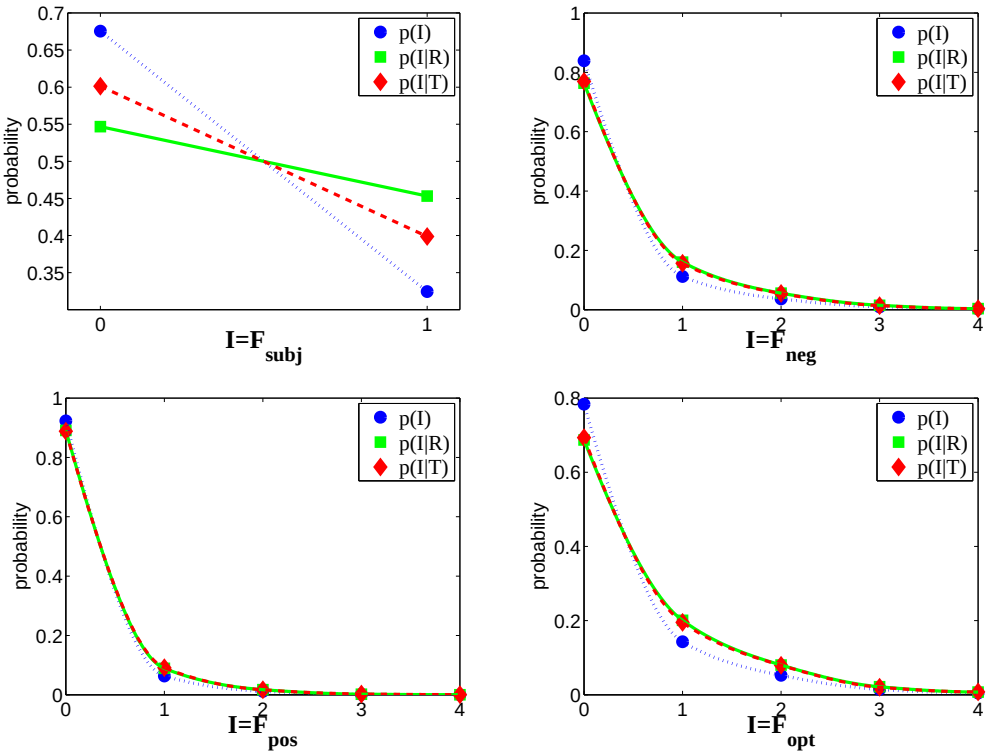


Figure 1.31: $p(I)$, $p(I|R)$ and $p(I|T)$ for the opinion-based features, given the 2S with $p(d|s)$ model.

cant improvements. The performance of the combined models is nearly the same as the performance of the context-based models alone and the few improvements found are tiny and, anyway, statistically insignificant. Thus, the incorporation of these query-independent features is useless.

We also tested the incorporation of opinion-based features into the 2S model with $p(d|s)$ with the empirical methods proposed above in this chapter. We considered values of w ranged from 1 to 99, in steps of 1. The performance after the incorporation of the opinion-based features given the empirical methods is shown in Table 1.22 and Figure 1.35.

In general, performance decreases after incorporating F_{pos} into 2S with $p(d|s)$. For the other features, improvements with respect to the baseline are obtained, especially with F_{subj} with the accuracy classifier. However, statistically significant improvements are never obtained. These outcomes are consistent with the analysis done before (Figure 1.31), where we anticipated that there might be little room for improvement.

Summing up, the gains obtained with opinion-based features on the top of the model 2S with $p(d|s)$ are modest and statistically insignificant. If

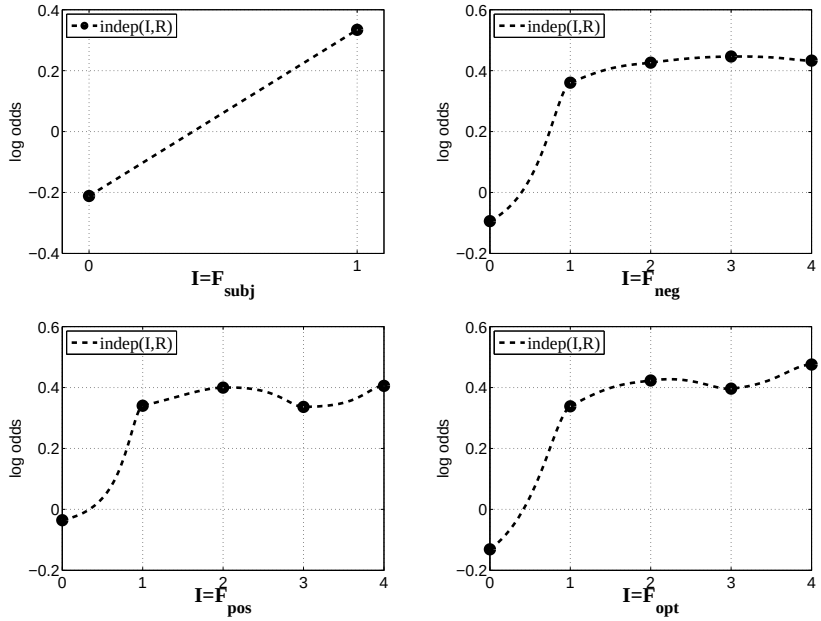


Figure 1.32: Adjustment under independence assumption given different opinion-based features.

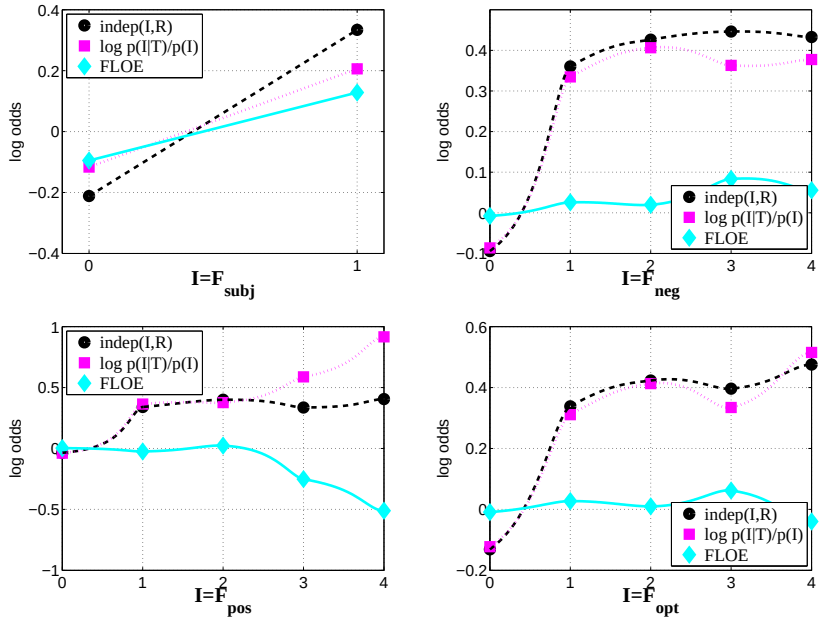


Figure 1.33: FLOE adjustment for the opinion-based features given the 2S with $p(d|s)$ model. Adding FLOE to $\log \frac{p(I|T)}{p(I)}$ we get the ideal adjustment, indep.

	2S with $p(d\|s)$ (baseline)	**F_{subj} (accuracy classifier)**	**F_{subj} (precision classifier)**	**F_{neg}**	**F_{pos}**	**F_{opt}**
			TREC 2003			
P@10	**0.7440**	**0.7440**	0.7400	**0.7440**	**0.7440**	0.7400
Δ%		$(+0.00)$	(-0.54)	$(+0.00)$	$(+0.00)$	(-0.54)
MAP	0.4117	**0.4119**	0.4114	0.4118	0.4117	0.4116
Δ%		$(+0.05)$	(-0.07)	$(+0.02)$	$(+0.00)$	(-0.02)
			TREC 2004			
P@10	0.4420	0.4420	0.4420	0.4420	**0.4440**	0.4420
Δ%		$(+0.00)$	$(+0.00)$	$(+0.00)$	$(+0.45)$	$(+0.00)$
MAP	0.2538	**0.2540**	0.2536	0.2538	**0.2540**	0.2536
Δ%		$(+0.08)$	(-0.08)	$(+0.00)$	$(+0.08)$	(-0.08)

Table 1.21: Retrieval performance of 2S with $p(d|s)$ and 2S with $p(d|s)$ + FLOE in the test collections given the opinion-based features.

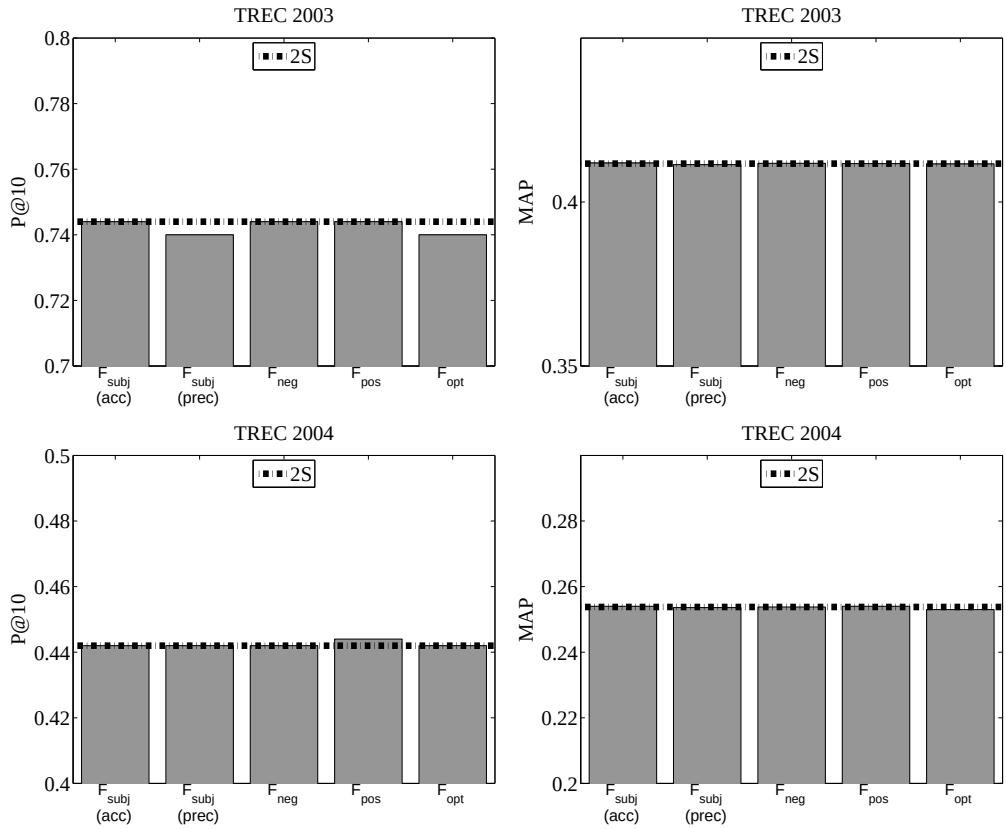


Figure 1.34: Retrieval performance of 2S with $p(d|s)$ and 2S with $p(d|s)$ + FLOE in the test collections given the opinion-based features.

we analyze comparatively the closeness between $p(I|R)$ and $p(I|T)$ for this model and compare it against a context-unaware model (tfisf), the difference becomes apparent (see Figure 1.36): 2S with $p(d|s)$ retrieves more subjective sentences than tfisf. To further understand why the model 2S with $p(d|s)$ al-

			2S with $p(d\|s)$+function(I)				
		2S with $p(d\|s)$ (baseline)	F_{subj} (accuracy classifier)	F_{subj} (precision classifier)	F_{neg}	F_{pos}	F_{opt}
		test: TREC 2003 (train: TREC 2004)					
	log				.7540	.7380	.7520
	Δ%				(+1.34)	(−0.81)	(+1.08)
	w				23	3	25
	linear	.7440	**.7840**	.7580	.7520	.7380	.7440
P@10	Δ%		(+5.38)	(+1.88)	(+1.08)	(−0.81)	(+0.00)
	w		76	66	9	2	1
	step				.7500	.7260	.7440
	Δ%				(+0.81)	(−2.42)	(+0.00)
	w				66	6	3
	log				.4166	.4121	.4176
	Δ%				(+1.19)	(+0.10)	(+1.43)
	w				35	3	10
	linear	.4117	**.4265**	.4172	.4164	.4118	.4165
MAP	Δ%		(+3.59)	(+1.34)	(+1.14)	(+0.02)	(+1.17)
	w		80	26	16	1	2
	step				.4168	.4118	.4171
	Δ%				(+1.24)	(+0.02)	(+1.31)
	w				25	1	24
		test: TREC 2004 (train: TREC 2003)					
	log				.4560	.4400	.4440
	Δ%				(+3.17)	(−0.45)	(+0.45)
	w				16	1	5
	linear	.4420	**.5240**	.5020	.4600	.4420	.4660
P@10	Δ%		(+18.55)	(+13.57)	(+4.07)	(+0.00)	(+5.43)
	w		59	99	8	1	1
	step				.4340	.4420	.4320
	Δ%				(−1.81)	(+0.00)	(−2.26)
	w				26	1	23
	log				.2584	.2508	.2584
	Δ%				(1.81)	(−1.18)	(+1.81)
	w				24	33	11
	linear	.2538	**.2730**	.2569	.2593	.2508	.2589
MAP	Δ%		(+7.57)	(+1.22)	(+2.17)	(−1.18)	(+2.01)
	w		52	28	16	18	2
	step				.2583	.2518	.2563
	Δ%				(+1.77)	(−0.79)	(+0.99)
	w				27	23	27

Table 1.22: Retrieval performance of 2S with $p(d|s)$ and 2S with $p(d|s)$ + function(I) in the test collections given the opinion-based features.

ready promotes subjective material, we report in Table 1.23 and Figure 1.37 the average and median $p(d|s)$ for the subjective and non-subjective sen-

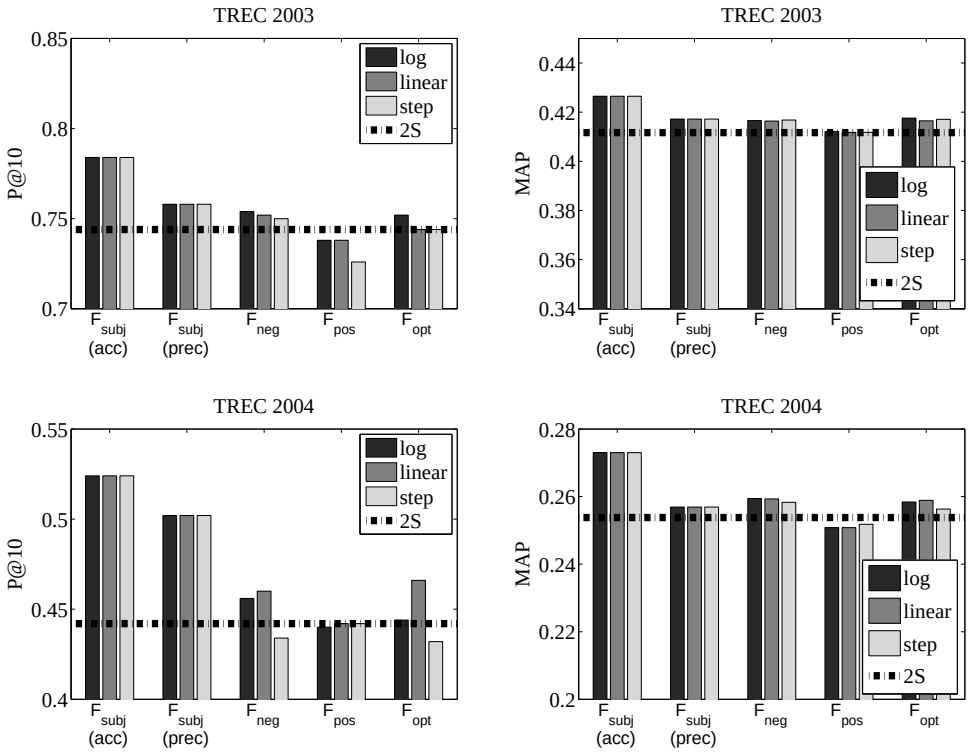


Figure 1.35: P@10 and MAP of 2S with $p(d|s)$ and 2S with $p(d|s)$ + function(I) in the test collections given the opinion-based features.

	avg $p(d\|s)$	**median $p(d\|s)$**
	TREC 2003	
$\boldsymbol{F_{subj} = 0}$	42.64	38.29
$\boldsymbol{F_{subj} = 1}$	50.02	47.48
	TREC 2004	
$\boldsymbol{F_{subj} = 0}$	43.78	40.01
$\boldsymbol{F_{subj} = 1}$	50.42	48.02

Table 1.23: Average and median $p(d|s)$ for the non-subjective and subjective sentences in TREC 2003 and TREC 2004 datasets.

tences in the collection. Given these statistics, it is obvious that subjective sentences on average tend to have higher $p(d|s)$. Models with $p(d|s)$ promote "central" sentences which, given this analysis, are good summaries of the document that contain subjective views. Therefore, this demonstrates that the incorporation of F_{subj} feature on the top of the 2S model with $p(d|s)$ does not give further benefits because this promotion of subjective sentences is implicitly captured by $p(d|s)$.

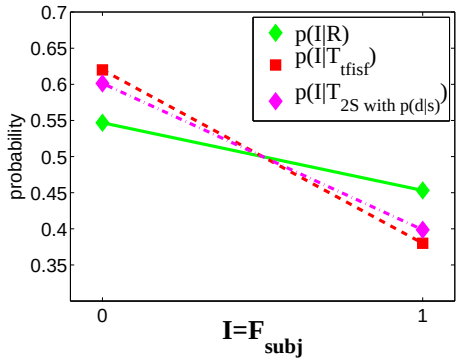


Figure 1.36: Comparison between $p(I|R)$, $p(I|T_{tfisf})$ and $p(I|T_{2S\ with\ p(d|s)})$ given the F_{subj} feature.

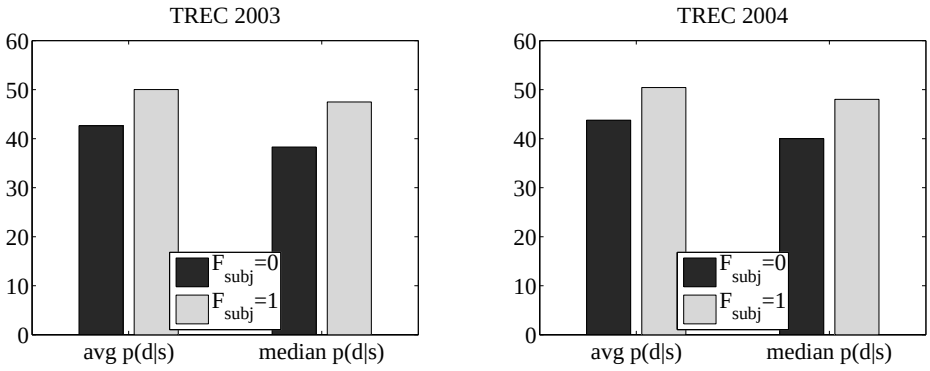


Figure 1.37: Average and median $p(d|s)$ for the non-subjective and subjective sentences in TREC 2003 and TREC 2004 datasets.

1.7 Conclusions

In this chapter we studied the impact of opinionated information on a sentence retrieval problem. We considered different opinion-based features and included them into high performing sentence retrieval models as query-independent weights. To this aim, we followed FLOE, a formal methodology to include properly query-independent evidence into existing models. Most of the models proposed in this chapter (either the ones derived directly from FLOE or those ones inspired by FLOE) outperform significantly a very competitive sentence retrieval baseline. We provided experimental evidence to show that the subjectivity of a sentence, the number of terms with negative orientation and the number of opinionated terms are sentence features that help to estimate relevance. Notably, the model that combines a regular baseline with the subjectivity of a sentence is very encouraging and boosts

performance (improvements from 9 to 26%). On the other hand, we also showed that sentence length can be combined with opinion-based features to further improve effectiveness. Named entity-based features were also tested but they led to negligible gains.

The use of opinion-based features in sentence retrieval is a novel contribution and the results reported here open up a new line of investigation: leveraging different forms of prior information in order to improve baseline retrieval. In this respect, in the future we will study different retrieval scenarios trying to understand when and why subjective content is more amenable to users. We believe that opinionated content might be valuable not only in sentence retrieval but also in other classical information retrieval problems.

In the second part of this chapter we proposed several novel probabilistic LMs to address the SR problem by including the local context. The context provided by the document means that the estimate of relevance for sentences is based on the sentence itself, the document that contains the sentence and the query. As part of the sentence Language Model, localized smoothing was included to provide a better estimate of the probability of a term in a sentence. The importance of sentences within the document was also included in our models. In a comprehensive set of experiments performed over several TREC test collections, we have compared the proposed models against existing SR models. Our experiments showed that using both forms of local context significantly outperforms the standard LM approach applied to sentence retrieval and the current state of the art sentence retrieval models. This is an important advancement in the development of effective SR methods. More specifically, it was found that:

- Using localized smoothing (2S-I) improves the performance of the LMs methods (by up to 13.8% improvement in MAP).
- Including sentence importance significantly improves the performance of all the LM approaches.
- LMs that use local context significantly outperform the current state of the art.

It was also shown that the improvements in the proposed methods were partly due to their tendency to favor longer sentences. This finding demonstrates that the naive application of document retrieval models to other retrieval tasks can lead to non-optimal performance; and warrants the development of sentence retrieval methods which account for the length normalization problem. These findings suggest that further progress in the area of sentence retrieval is possible, and that more sophisticated and more effective

models can be developed by incorporating the local context within the LM framework. This work motivates future research and development on:

(i) developing other methods in a principled fashion to also include local context, i.e. changing the vector representation in tfisf, including a sentence importance factor, or including the local context in the classic Probabilistic Model for IR,

(ii) instead of considering the closest surrounding sentences (previous and next), consider a variable number of surrounding sentences,

(iii) define a four-mixture model that combines the sentence, the local context, the document and the background model,

(iv) the modification of pivoted length normalization, [SBMM96] or BM25 to do SR promoting long sentences; or sentence priors for LMs to investigate the length normalization issues,

(v) other estimation methods of the LMs and priors, along with automatic parameter estimation techniques, and

(vi) the application and extension of the Language Modeling framework to other tasks, such as query-biased summarization or novelty detection.

Finally, we also combined sentence retrieval models that use the local context with opinion-based features. The incorporation of these features did not yield to higher performance with respect to the performance obtained with local context alone. This happens because the local context models are implicitly promoting opinionated information.

To sum up, we studied here how to improve standard sentence retrieval models with two different approaches: a) including the context into the estimation of relevance, and b) incorporating query-independent information. Both approaches improved significantly the original sentence retrieval methods, but their combination did not yield to additional gains. Among the different contextual methods proposed here, 2S with sentence importance and Dirichlet with $p(d|s)$ are methods that, generally, provide the highest P@10 and MAP, respectively. However, considering subjectivity as a query-independent feature and incorporating it into tfisf with a linear method is the approach that, overall, performs the highest.

Chapter 2

Novelty Detection

The goal of novelty detection is to provide the user with a list of sentences[1] that are relevant to the user's information need and, additionally, contain new information that has not been seen previously in the list. It is assumed that the user does not know anything about the topic at the time of the initial relevant document (sentence) and all learning happens in the order of document (sentence) retrieval, i.e. we assume that the user is most concerned about finding new information in the list of sentences and he/she is tolerant of reading information he/she already knows because of his/her background knowledge [SH03, Sob04, SH05].

In order to evaluate our novelty detection methods we considered the TREC 2002, 2003 and 2004 Novelty Tracks [Har02, SH03, Sob04]. These test collections supply relevance and novelty judgments at sentence level for each topic. Because the TREC 2002 contains a very low amount of relevant sentences (about the 2%) and nearly every relevant sentences was declared as novel (about the 91%), groups researching novelty detection methods were not encouraged to use that data for further experiments [SH05, Li06]. We therefore employ only TREC 2003 and TREC 2004 datasets in our evaluation.

In this chapter we study several novelty detection methods given two different initial situations: a) from a ranking of sentences sentences judged as relevant (perfect relevance), and b) from a ranking of *estimated* relevant sentences (non-perfect relevance). The former corresponds with the set of sentences that the TREC assessors judged as relevant for each query, preserving their original order. This perfect relevance scenario is precisely the configuration of Task 2 in TREC Novelty Tracks (given the relevant sentences

[1] In this thesis we deal with novelty detection at sentence level. As Soboroff and Harman indicated in [SH05], document-level novelty detection, while intuitive, is problematic because every document contains something new, particularly in the news domain.

in all documents, identify all novel sentences) and permits to study novelty detection without interferences coming from non-relevant material [Li06]. On the other hand, the non-perfect relevance situation starts from a ranked set of sentences that were estimated as relevant by a given sentence retrieval baseline (tfisf in our case). This situation in more realistic because, usually, we do not know what sentences are relevant. With non-perfect relevance, it has been shown that the effectiveness of novelty detection depends strongly on the quality of the initial sentence retrieval baseline [AWB03]. There is an issue with non-perfect relevance that affects to the quality of the evaluation. The judgment of novel sentences was based on a particular set of relevant sentences in a presumed order. It is not very accurate to evaluate a system's performance if the ranked sentences of a novelty detection system have a different order from the particular set. Still, the non-perfect relevance scenario has been a popular approach in the literature (as a matter of fact, it is inherent to Task 1 of the Novelty Tracks) and, therefore, we should consider it in our study.

2.1 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) [CG98]. 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, some novelty measures were proposed and evaluated [ZCL03]. More specifically, a cosine similarity measure and a redundancy measure based on a mixture of three Language Models were proposed.

Providing many relevant documents to the user is not usually desirable, especially in a real environment such as the web, where users do not tend to look beyond the first top-ranked documents. Chen and Karger [CK06] stated that attempting to retrieve many relevant documents can actually reduce the chances of finding any relevant documents and they proposed a framework to reduce the top retrieved documents. To this aim, they estimated the relevance of a document by assuming that all its previous documents are not relevant. Implicitly, this promotes diversity because it tries to retrieve documents that are relevant to the query but different to previous ones.

Wang and Zhu [WZ09] adopted the Portfolio Theory, an economy theory related to investment in the financial market, to document retrieval and stated that the most appropriate retrieval method will be the one that supplies the right combination of relevant documents in the top-ranked positions.

They argue that ranking under uncertainty is not just about picking individual relevant documents but about choosing the right combination of relevant documents. This resembles the investment problem in financial markets in the sense that, for instance, investors need to select the set of stocks (portfolio) that will provide the highest future benefits, bearing in mind the available investment budget.

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 Simple New Word Count, Set Difference and Cosine Distance [AWB03]. 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 demonstrated to work reasonably well, but they lack a formal modeling of the elements involved.

Language Models (LMs) are powerful tools that have been demonstrated to work well in IR [CL03]. Since the seminal proposal in the late 90's [PC98], 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 [AWB03] a LM is created for the current sentence and another LM is created for the set of previously seen sentences. The authors propose to obtain novelty scores by applying the divergence between both models. Additionally, in [Fer07], we used this model to study the impact of smoothing on novelty detection performance.

In [ZCL03] 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.

Li and Croft [LC08] defined the concept of novelty based on information patterns contained in sentences, such as named-entities, opinions, etc. they also proposed a mechanism uses such patterns in order to obtain an effective method to estimate novelty.

In the following sections we propose several methods to address the novelty detection problem. These methods are either modifications of existing mechanisms or new proposals to address the problem in a different way. First, Local Context Analysis [XC96] is used to define a query-oriented vocabulary that is applied to drive novelty detection [FL07]. To this aim, we modify current state of the art novelty detection methods [AWB03] to incorporate such vocabulary. This helps to avoid redundant sentences and it is particularly useful as a high precision mechanism. Next, we study the performance of different LM-based models and propose an effective variant for some of them. This modification is not only more efficient but also preserves (or slightly increases) the effectiveness of the original models. We also propose and eval-

uate an approach based on a mixture model, similar to the model proposed in [ZCL03], that employs the Expectation-Maximization algorithm in order to estimate novelty. The aim here is to develop a formal parameter-free mechanism that addresses effectively the novelty detection problem. However, we demonstrate that this method does not perform effectively for novelty detection. Finally, after analyzing these approaches against two different scenarios (perfect and non-perfect relevance), we propose new novelty detection methods that further improve performance. These methods are based on freezing the early positions of the rank and estimating novelty starting from lower positions. This is done here following a query-independent or a query-dependent approach.

2.2 Novelty Detection with Non-Perfect Relevance

The methods proposed to address the novelty task given the non-perfect relevance ranking involve two stages: a) *sentence retrieval*: we use a standard sentence retrieval method in order to obtain a ranking of estimated relevant sentences, and b) *novelty detection*: we re-order sentences according to a novelty/redundancy criterion. In order to compute novelty, it is not feasible to re-order all sentences provided by a sentence retrieval method because there may exist a large amount of estimated relevant sentences. Therefore, it is usual to prune the ranked set of estimated relevant sentences and consider only the set of top-ranked sentences. In our case, we consider the top 10% of estimated relevant sentences[2].

Given a ranked set of estimated relevant sentences (obtained with tfisf), we consider two different sets of estimated relevant sentences as our novelty baselines: a) **BNN** (Baseline with No Novelty detection), which ranks sentences using directly its similarity score (from the tfisf baseline [AWB03]); and b) **BDOC** (Baseline ordered by DOCument), which consists of a re-ordering of the sentences from the BNN ranking where 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. Both baselines have been used in the past [AWB03, Li06, Fer07] but there is not any comparative study evaluating its relative merits for novelty detection.

[2]Allan et al. followed the same approach in [AWB03]. In [TTC10], the authors studied the behavior of several metrics and considered a range of thresholds, i.e. given a ranked set of sentences, they studied the impact of the number of top-ranked sentences that enter in the novelty detection module.

	BNN	BDOC
	TREC 2003	
P@10	.5300	.5660
$\Delta\%$		$(+6.79)$
MAP	.1012	.1053
$\Delta\%$		$(+4.05)$
	TREC 2004	
P@10	.2080	.2540
$\Delta\%$		$(+22.12)$
MAP	.0527	.0632*
$\Delta\%$		$(+19.92)$

Table 2.1: Performance of BNN and BDOC baselines.

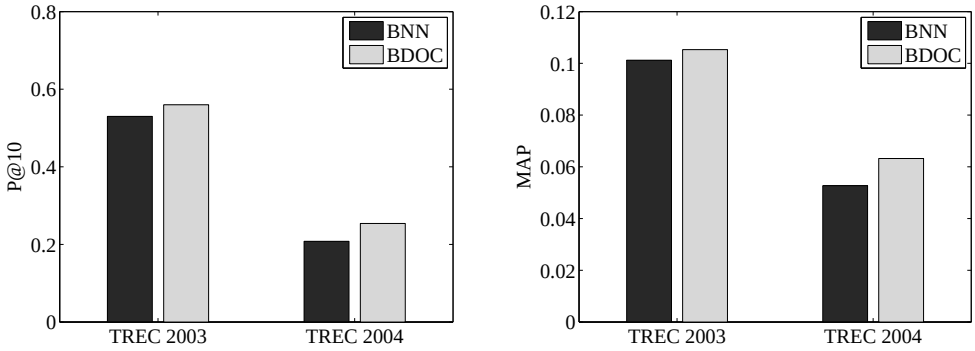


Figure 2.1: Comparison between BNN and BDOC performance given a non-perfect relevance ranking.

2.2.1 Performance of the Novelty Baselines

In this section we compare the performance of BNN and BDOC baselines. Figure 2.1 and Table 2.1 report P@10 and MAP for these baselines in the test collections. BDOC performs better than BNN with any of the TREC novelty datasets. BDOC preserves the natural order of sentences within documents, and takes documents in the order given by the task. Sentences within documents, sorted by their natural order, do not tend to repeat similar information. In contrast, when sentences are sorted by their similarity score (BNN), similar content is likely to be presented by consecutive sentences. This is particularly apparent with P@10: BDOC supplies 10 sentences that, overall, are less redundant than the top 10 sentences supplied by BNN, which is solely focused on promoting sentences that are relevant to the query. BDOC is, therefore, a basic way to move from a relevance-oriented

rank to a rank that promotes diversity (by taking sentences in their natural order from documents considered in the order given by NIST).

Given the results shown above, BDOC is clearly more effective than BNN and, therefore, we will consider BDOC as our reference baseline.

2.2.2 Novelty Detection: Preliminaries

Any novelty detection method starts from a ranked set of sentences and aims at producing a new ranking of sentences (in decreasing order of novelty). Standard methods start often from a relevance ranking re-ordered in a BDOC fashion and proceed as follows. For each query, the first sentence of the ranking is always estimated as novel (regardless of the novelty detection method) because, initially, the user is assumed to know nothing (everything is novel). Therefore, a maximum novelty score is assigned to the first sentence. The estimation of novelty for the remaining sentences in the ranking is dependent on the specific novelty detection method that we use. In the next subsections we present different novelty detection techniques adapted to this scenario.

In Section 2.2.3 we analyze the performance of well-known novelty detection methods. Section 2.2.3.1 proposes a variant of these methods consisting of normalizing novelty scores by sentence length. In Section 2.2.4 we study the impact of using a query-based vocabulary to refine the novelty detection process by considering only on-topic terms. In Section 2.2.5 we propose novelty detection methods based on Language Models. Finally, in Section 2.2.6 we combine different Language Models in order to estimate automatically novelty scores by applying the Expectation-Maximization algorithm.

2.2.3 Standard Novelty Detection Methods

Given a ranked set of sentences (BDOC in our case), some methods have been proposed in the literature to compute the overlapping between each sentence and the previously seen sentences. We have chosen three methods which are simple and robust [AWB03]: Simple New Word Count (NewWords), Set Difference (SetDif) and Cosine Distance (CosDist).

NewWords counts the number of words contained in the current sentence, s_i, that have not been seen in any previous sentence:

$$N_{nw}(s_i|s_1, ..., s_{i-1}) = \left| W_{s_i} \cap \overline{\bigcup_{j=1}^{i-1} W_{s_j}} \right| \tag{2.1}$$

where W_{s_i} is the set of words appearing in the sentence s_i.

SetDif computes the number of different words between each sentence s_i and the previously seen sentence that is the most similar to s_i:

$$N_{sd}(s_i|s_1, ..., s_{i-1}) = \min_{1 \leq j \leq i-1} N_{sd}(s_i|s_j) \tag{2.2}$$

$$N_{sd}(s_i|s_j) = \left|W_{s_i} \cap \overline{W_{s_j}}\right|$$

Representing a sentence as a vector in a m-dimensional space (where m is the number of terms in the vocabulary and the weights on individual dimensions are determined by a term weighting function), **CosDist** computes the novelty score of a sentence as the negative of the cosine of the angle between a sentence vector and the most similar previously seen sentence, i.e.:

$$N_{cd}(s_i|s_1, ..., s_{i-1}) = \min_{1 \leq j \leq i-1} N_{cd}(s_i|s_j) \tag{2.3}$$

$$N_{cd}(s_i|s_j) = -\frac{\sum_{k=1}^{m} w_k(s_i) \cdot w_k(s_j)}{\sqrt{\sum_{k=1}^{m} w_k(s_i)^2 \cdot \sum_{k=1}^{m} w_k(s_j)^2}}$$

where $w_k(s_i)$ is the weight for word w_k in sentence s_i. In this case, the weighting function (w_k) can be defined as follows:

$$w_k(s_i) = \frac{c(w_k, s_i)}{c(w_k, s_i) + 0.5 + (1.5 \cdot \frac{c(s_i)}{asl})} \cdot \frac{\log \frac{N+0.5}{sf(w_k)}}{\log(N + 1.0)} \tag{2.4}$$

where asl is the average number of words in a presumed relevant sentence for the topic, $sf(w_k)$ is the number of presumed relevant sentences for the topic that contain the word w_k, N is the number of presumed relevant sentences for the topic, $c(w_k, s_i)$ is the frequency of w_k within the sentence s_i and $c(s_i)$ is the number of terms that s_i contains.

In Table 2.2 and Figure 2.2 we report the performance of these novelty detection methods in the TREC 2003 and TREC 2004 novelty datasets. The best result for each metric and collection is bolded.

Note that, in all cases, NewWords performs better than SetDif and CosDist. In TREC 2003 the differences between NewWords and the baseline are statistically significant. Although NewWords' performance is higher than the performance of the baseline in TREC 2004, the differences are not statistically significant. SetDif and the baseline perform roughly the same while CosDist, usually, performs statistically worse than the baseline. Therefore, CosDist does not work well for the novelty detection problem (at least, given our non-perfect relevance ranking). Observe that CosDist incorporates tf-idf

	BDOC (baseline)	NewWords	SetDif	CosDist
		TREC 2003		
P@10	.5660	**.6380***†	.6100	.5000*
$\Delta\%$		$(+12.72)$	$(+8.93)$	(-11.66)
MAP	.1053	**.1182***†	.1169*†	.1042
$\Delta\%$		$(+12.25)$	$(+11.02)$	(-1.04)
		TREC 2004		
P@10	.2540	**.2800**	.2720	.1920*†
$\Delta\%$		$(+10.24)$	$(+7.09)$	(-24.41)
MAP	.0632	**.0677**	.0670	.0549*†
$\Delta\%$		$(+7.12)$	$(+6.01)$	(-13.13)

Table 2.2: NewWords, SetDif and CosDist performance against the BDOC baseline.

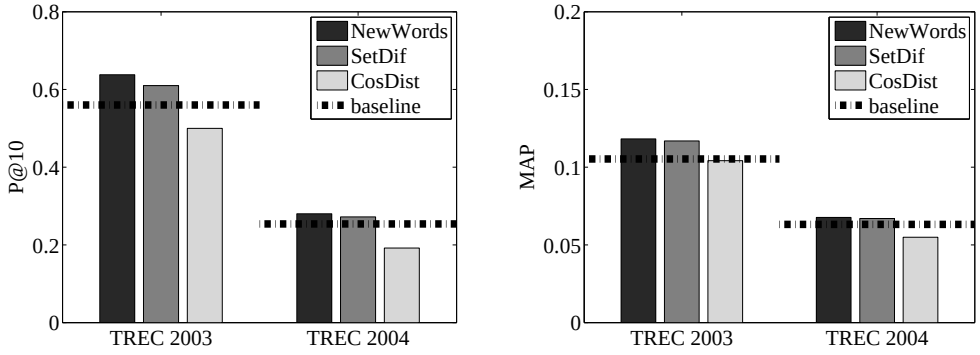


Figure 2.2: Comparison between the BDOC baseline and standard novelty detection methods.

weights while SetDif/NewWords are based on raw term counts. Still, SetDif/NewWords perform better than CosDist. This seems to indicate that evolved term weighting is not needed for novelty detection purposes.

Sentences ranked high by NewWords will have many terms that are not in common with any of the previously seen sentences. On the other hand, SetDif and CosDist make a sentence-to-sentence comparison between the current sentence and the previous ones. With these two methods, the final novelty score depends directly on the most similar sentence in the history. Therefore, a sentence might be totally redundant (e.g. because its contents are covered by two different sentences in the history) but SetDif or CosDist estimate its degree of redundancy focusing only on the sentence components that overlap with the most similar sentence in the history. In this respect,

s_1: today it is warm
s_2: John is wearing a coat
s_3: although it is warm today, John is wearing a coat

$N_{nw}(s_3|s_1, s_2)$: 0; $N_{sd}(s_3|s_1, s_2)$: 2; $N_{cd}(s_3|s_1, s_2)$: -0.77

Figure 2.3: An example of how state of the art novelty detection methods compute novelty scores. Note that stopwords are removed (we mark them with a light gray color).

NewWords seems to be more robust because the sentence terms are *classified* as redundant provided that they appeared somewhere before. For instance, given the three sentences s_1, s_2 and s_3, shown in Figure 2.3 (we assume that all of them are relevant for a given topic), s_3 should not be considered novel because it does not provide additional information with respect to s_1 and s_2. In fact, NewWords assigns a null novelty score to this sentence. However, SetDif and CosDist do not identify this sentence as fully redundant because, as explained above, s_3's novelty score comes from the most similar previous sentence (in this case, s_2)[3].

Note that CosDist is a symmetric measure, i.e. given a sentence s_i and a previous sentence s_j, $N_{cd}(s_i|s_j) = N_{cd}(s_j|s_i)$. This happens because CosDist considers novelty as the opposite to redundancy. In fact, it first computes the similarity between s_i and s_j (the similarity is always a symmetric feature) and, next, subtracts to the maximum novelty score (0, in this case) the similarity value and assigns it to the sentence s_i. Nevertheless, the ordering of sentences is important in order to compute novelty [TTC10]. For instance, given the sentences from Figure 2.4, s_2 is likely to be predicted as novel due to the low similarity with respect to s_1. However, s_2 does not provide additional information and, therefore, it should be estimated as non-novel.

s_1: I was unable to rent a car because there are not cars left
s_2: There are not cars left

Figure 2.4: An example to explain the behavior of symmetry and asymmetry. Note that stopwords are removed (we mark them with a light gray color).

In contrast, NewWords and SetDif are asymmetric measures. A deep

[3]With CosDist the range of scores is $[-1, 0]$: obtaining scores closed to 0 indicates that two sentences s_i and s_j are very different (novel). However, if we obtain scores closed to -1, sentences s_i and s_j are very similar (redundant).

study of symmetric and asymmetric measures and their behavior has been published in [TTC10]. We show here that CosDist, a symmetric measure, performs worse than NewWords and SetDif (asymmetric measures) in terms of P@10 and MAP. In fact, in [TTC10] the authors reported that symmetric metrics are only useful in high recall scenarios with low redundancy in the initial ranked set.

2.2.3.1 Normalizing Standard Novelty Detection Methods

The novelty detection methods explained in the previous section do not consider explicitly the length of sentences at novelty score calculation time. Depending on the novelty method utilized, the length of a sentence might have a different effect on the novelty estimation trends. We therefore felt that it was interesting to study length-based variations of the models described above. In this section we propose a variant of the novelty methods that normalizes the novelty scores by dividing by the number of terms in the sentences (sentence length). In this way, we promote sentences that are novel but not too verbose.

In Table 2.3 and Figure 2.5 we show a comparison between NewWords, SetDif and CosDist and their corresponding normalized variants (NewWords$_n$, SetDif$_n$ and CosDist$_n$). These results indicate that this variant is only helpful for CosDist. CosDist$_n$ provides statistically significant differences with respect to CosDist[4]. In contrast, the variants of the rest of methods (NewWords$_n$ and SetDif$_n$) do not outperform their original versions. This is particularly noticeable with NewWords$_n$, which is significantly worse than NewWords. To understand these results we first present in Table 2.4 the average sentence length for each TREC Novelty dataset (first column), the average length of relevant (second column) and novel sentences (third column), and the average length of sentences that are relevant but not novel (fourth column). Observe that, on average, novel sentences are longer than relevant sentences, and relevant sentences are longer than sentences in the collection. Therefore, promoting longer sentences might be a way to increase the performance of novelty detection methods.

NewWords and SetDif count the number of terms that are present in the current sentence s_i but were not seen before (either in any previous sentence - NewWords - or in the most similar sentence - SetDif -). Therefore, the longer s_i is, the more likely s_i contains unseen terms. Consequently, NewWords and SetDif scores grow as sentence length increases. Since NewWords$_n$ and SetDif$_n$ normalize by sentence length they retrieve shorter sentences on

[4]However, CosDist$_n$ is still unable to outperform the BDOC baseline.

	NewWords	NewWords$_n$	SetDif	SetDif$_n$	CosDist	CosDist$_n$
			TREC 2003			
P@10	**.6380**	.5820*†	**.6100**	.6060	.5000	**.5600***†
$\Delta\%$		(-8.78)		(-0.66)		$(+12.00)$
MAP	**.1182**	.1111*†	**.1169**	.1145	.1042	**.1103***†
$\Delta\%$		(-6.01)		(-2.05)		$(+5.85)$
			TREC 2004			
P@10	**.2800**	.2520	**.2720**	.2320*	.1920	**.2140***†
$\Delta\%$		(-10.00)		(-14.71)		$(+11.46)$
MAP	**.0677**	.0622*	**.0670**	.0625*	.0549	**.0596***†
$\Delta\%$		(-8.12)		(-6.72)		$(+8.56)$

Table 2.3: Performance of the standard novelty detection methods and their normalized variants.

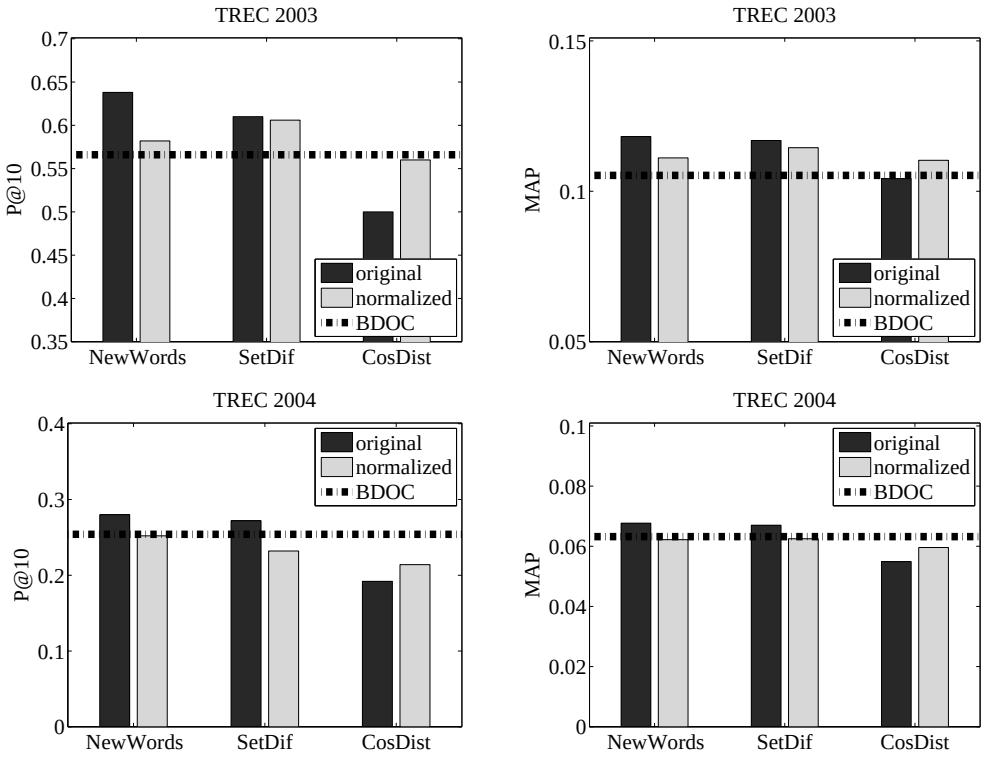


Figure 2.5: Performance of the standard methods and their normalized variants.

average. As anticipated by Table 2.4 and confirmed by results in Table 2.3, this harms performance.

With CosDist, novelty scores are negative and, therefore, the effect of length normalization is the opposite: longer sentences are promoted in the ranking of novelty. This explains why CosDist$_n$ works better than CosDist.

	Collection	Relevant	Novel	Relevant $\cap \overline{Novel}$
TREC 2003	9.92	12.68	12.89	12.25
TREC 2004	9.66	13.30	13.81	12.93

Table 2.4: Average sentence length in each TREC novelty dataset, for the set of relevant sentences, the set of novel sentences and the set of relevant sentences that are not novel.

2.2.4 Novelty Detection Based on Vocabulary Pruning

Some researchers have proposed that the estimation of novelty for a given sentence s_i should be based on the set of seen sentences that share common meanings with s_i [ZZM06]. In this way, the degree of redundancy of such sentence is not influenced by past sentences that are totally unrelated. The intuition is that sentences that are off-topic should not be used to classify new sentences as novel or redundant. The user perceives novelty with respect to his/her topic of interest and, therefore, our novelty detection estimations should only be based on the information related to the topic that has been acquired during his/her interaction with the system. Observe also that we work here from a non-perfect relevance ranking and, thus, off-topic material might severely affect novelty detection performance.

We propose to retrieve relevant and novel sentences using a vocabulary that helps us to focus the estimation of novelty on query-related terms. Novelty estimation might be more robust if focused on this set of terms. In order to extract this vocabulary, two mechanisms are considered: Local Context Analysis (LCA)[XC00] and Divergence From Randomness (DFR)[HO07]. These methods have been successfully applied in different information retrieval areas. We check here whether or not these mechanisms are useful to drive the selection of novel material.

2.2.4.1 Local Context Analysis

Local Context Analysis (LCA) is a method based on the idea that a common term from the top-ranked relevant documents (or passages) will tend to co-occur with query terms within the top-ranked documents (or passages) [XC00]. This technique has been successfully applied in different areas, such as query expansion [XC00, XC96].

We apply here LCA to produce a set of query-related terms, and the novelty scores are adjusted accordingly. Given the initial ranking, the importance of the terms in the top ranked sentences is computed adopting the expression from [XC96]:

$$bel(q,t) = \prod_{t_i \in q} \left(\delta + \frac{\log(af(t,t_i)) \cdot idf(t)}{\log(N)} \right)^{idf(t_i)} \tag{2.5}$$

where t is a term, δ is 0.1 (a constant) to avoid zero *bel* value, $af(t,t_i) = \sum_{j=1}^{n} c(t_i,s_j) \cdot c(t,s_j)$, where $c(t_i,s_j)$ is the number of occurrences of the term t_i in the sentence s_j and $c(t,s_j)$ is the number of occurrences of the term t in the sentence s_j; $idf(x) = \min\left(1.0, \frac{\log_{10}(\frac{N}{N_x})}{5.0}\right)$, where N_x is the number of sentences containing the term x, and N is the number of sentences in the collection.

This measure can be applied to rank terms in decreasing order of estimated importance given a query. Selecting the top ranked terms we can conform a query-oriented vocabulary (T_q). Using this vocabulary, we compute variations of NewWords, SetDif and CosDist for each sentence as follows:

$$N_{nw_{LCA}}(s_i|s_1,...,s_{i-1}) = \left| W_{LCA_{s_i,q}} \cap \overline{\bigcup_{j=1}^{i-1} W_{LCA_{s_j,q}}} \right| \tag{2.6}$$

,

$$N_{sd_{LCA}}(s_i|s_1,...,s_{i-1}) = \min_{1 \le j \le i-1} N_{sd_{LCA}}(s_i|s_j) \tag{2.7}$$

$$N_{sd_{LCA}}(s_i|s_j) = \left| W_{LCA_{s_i,q}} \cap \overline{W_{LCA_{s_j,q}}} \right|$$

, and

$$N_{cd_{LCA}}(s_i|s_1,...,s_{i-1}) = \min_{1 \le j \le i-1} N_{cd}(s_i|s_j) \tag{2.8}$$

$$N_{cd}(s_i|s_j) = -\frac{\sum_{k=1}^{m} w_k(s_i) \cdot w_k(s_j)}{\sqrt{\sum_{k=1}^{m} w_k(s_i)^2 \cdot \sum_{k=1}^{m} w_k(s_j)^2}}$$

$$w_k(s_i) = \begin{cases} \frac{c(w_k,s_i)}{c(w_k,s_i)+0.5+(1.5 \cdot \frac{c(s_i)}{asl})} \cdot \frac{\log \frac{N+0.5}{sf(w_k)}}{\log(N+1.0)} & \text{, if } w_k \in W_{LCA_{s_i,q}} \\ 0 & \text{, otherwise} \end{cases}$$

where $W_{LCA_{s,q}} = W_s \cap T_q$, and W_s is the set of terms in the sentence s. These measures are variants of the original novelty detection measures where the terms taken into account in the sentences are only those from T_q. This can be seen as a vocabulary pruning method that *reduces* sentences to their query-related parts.

We applied LCA considering the top terms (highest $bel(q,t)$) given the top sentences extracted from the baseline. We ran some preliminary experiments and concluded that the top 25 sentences is a proper configuration to

extract important terms. Figure 2.6 shows the performance of the variants proposed here considering different vocabulary sizes[5]. The Figure plots also the performance of the original methods and the performance of the BDOC baseline.

The main conclusion here is that the larger the vocabulary, the higher performance. Our results indicate that precision at top ranks might be further improved if redundancy decisions are made in terms of a more focused vocabulary. Nevertheless, MAP results are less consistent. This seems to indicate that LCA-based pruning is a high precision strategy that may not lead to improvements in MAP.

We found no significant differences in terms of performance between vocabulary sizes of 500 and 1000 terms. Usually, the total number of unique terms in the top 25 sentences is less than 500 and, therefore, setting $|T_q|$ to either 500 or 1000 makes that we select *all* terms from the top-25 sentences. This means that, given our current results, a simple method (based on extracting all terms appearing in the top 25 sentences) would perform well and would not require LCA.

In the next subsection we apply an alternative term selection approach based on the informativeness of terms within sentences.

2.2.4.2 Divergence From Randomness

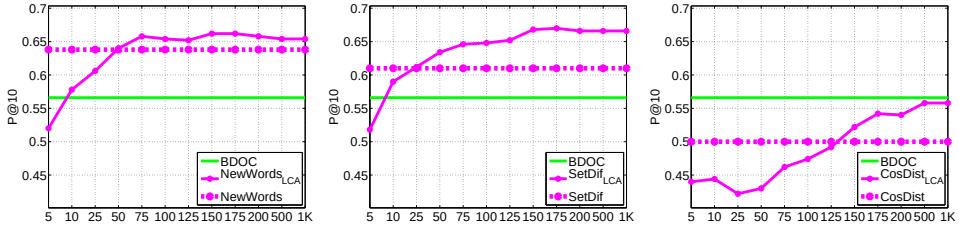
Divergence From Randomness (DFR) measures the informativeness of a term in a document (or passage) through the divergence of the terms' distribution in a document and a random distribution [AJR02, Ama03]. As we did in the LCA case, we consider here this framework to estimate the most important terms in the top ranked sentences, i.e. to get a vocabulary T_q. To extract this vocabulary, we weight all terms in the top ranked sentences by considering the Bol model that uses Bose-Einstein statistics [Ama03, MHPO04, MHPO05] because it has been proved to be the most effective DFR term weighting model [HO07]. This model weights terms t as follows:

$$w(t) = c_t \cdot \log_2 \frac{1 + P_t}{P_t} + \log_2(1 + P_t) \tag{2.9}$$

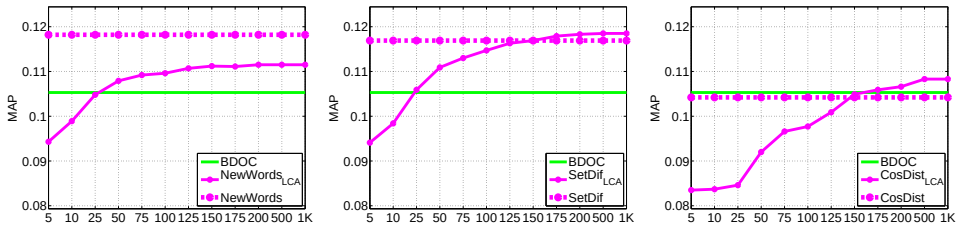
where c_t is the frequency of t in the top-ranked sentences and P_t is given by $\frac{c(t)}{N}$ (where $c(t)$ is the frequency of t in the collection and N is the number of sentences in the collection).

Next, we build the vocabulary T_q by extracting terms with the highest weights. The novelty detection mechanisms, given T_q, are analogous to the

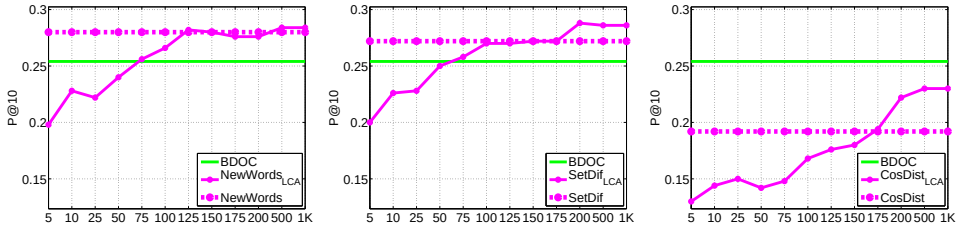
[5]We made experiments with these vocabulary sizes: 5, 10, 25, 50, 75, 100, 125, 150, 175, 200, 500, 1000.



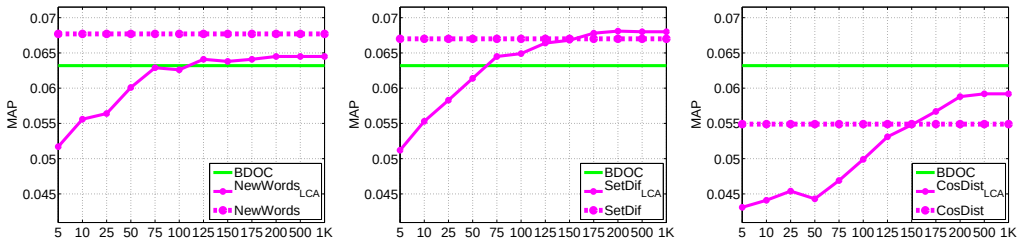
(a) P@10 TREC 2003



(b) MAP TREC 2003



(c) P@10 TREC 2004

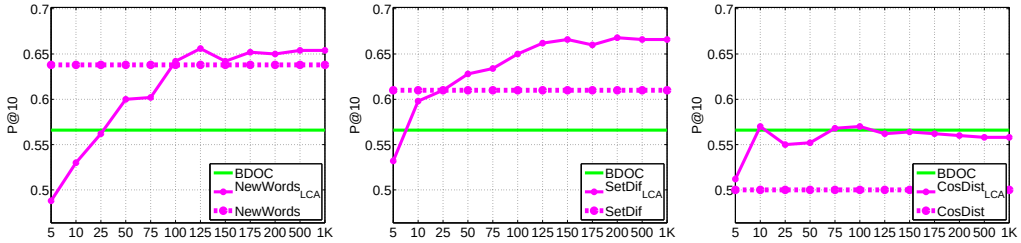


(d) MAP TREC 2004

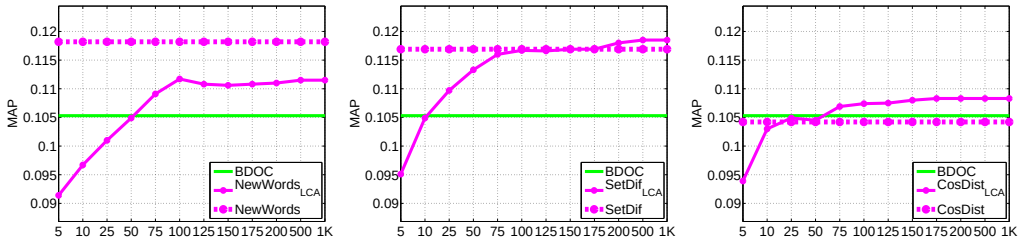
Figure 2.6: Results of state of the art novelty methods using LCA-based vocabulary pruning.

ones presented in the previous section. The number of sentences used to extract these terms was also fixed to 25. Figure 2.7 shows the performance of the novelty detection methods considering the vocabulary extracted with DFR.

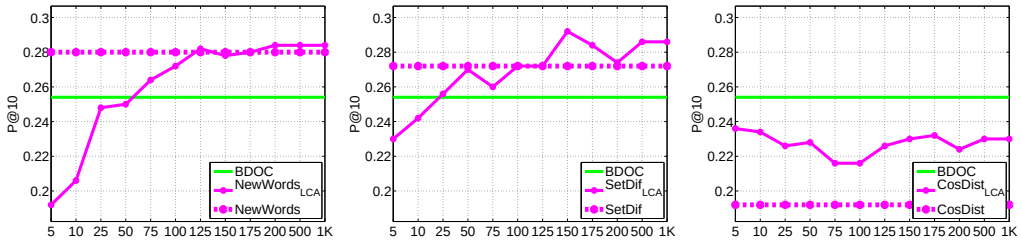
Again, the best performance is reached with large vocabularies. Although



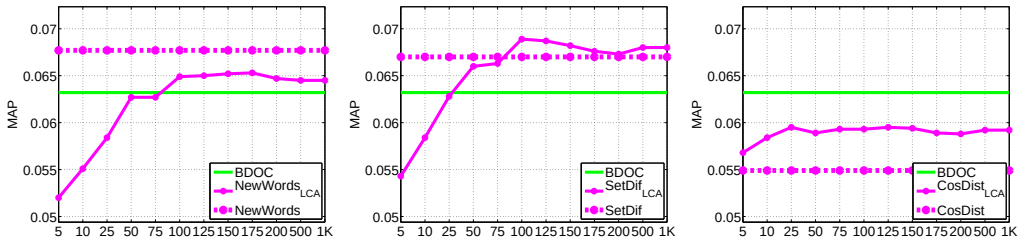
(a) P@10 TREC 2003



(b) MAP TREC 2003



(c) P@10 TREC 2004



(d) MAP TREC 2004

Figure 2.7: Results of state of the art novelty methods using DFR-based vocabulary pruning.

there are some exceptions, the best performance tends to be achieved when all terms in the top-ranked sentences are considered as a vocabulary.

These results further support the conclusions extracted with LCA: a simple method, based on extracting all terms appearing in the top 25 sentences,

	NewWords		SetDif		CosDist	
	original	**vocab.**	**original**	**vocab.**	**original**	**vocab.**
			TREC 2003			
P@10	.6380	**.6540**	.6100	**.6660***	.5000	**.5580***†
$\Delta\%$		(+*2.50*)		(+*9.18*)		(+*11.60*)
MAP	**.1182**	.1115*	.1169	**.1185**	.1042	**.1083***
$\Delta\%$		(−*5.67*)		(+*1.34*)		(+*3.93*)
			TREC 2004			
P@10	.2800	**.2840**	.2720	**.2860**	.1920	**.2300***†
$\Delta\%$		(+*1.43*)		(+*5.15*)		(+*19.79*)
MAP	**.0677**	.0645	.0670	**.0680**	.0549	**.0592***†
$\Delta\%$		(−*4.73*)		(+*1.49*)		(+*7.83*)

Table 2.5: Comparison of performance between standard novelty detection methods and the variant that uses vocabulary pruning (vocabulary composed of all terms in top-25 retrieved sentences).

performs well and more evolved term selection methods, such as DFR or LCA, are not needed.

Finally, the performance of this simple novelty detection method based on vocabulary pruning is compared against the original methods in Table 2.5 and Figure 2.8. CosDist with vocabulary pruning outperforms significantly the standard CosDist. However, with the other novelty detection methods, the performance tends to be higher when the vocabulary is large but, usually, the difference is not statistically significant with respect to the standard methods. Overall, the results show that using vocabulary pruning is good when we need to retrieve top 10 novel sentences. However, with MAP the vocabulary-based pruning approach harms NewWords but is somehow beneficial for SetDif and CosDist.

So far we have studied the performance of state of the art novelty detection methods and proposed variations of these novelty detection methods: a sentence-length normalization and a vocabulary pruning approach based on guiding the novelty detection towards on-topic terms. However, these methods are somehow ad-hoc. In the next section we study formal methods based on Language Models to address the novelty detection process. In later sections, a comparison between the standard novelty methods explained so far (ad-hoc methods) and formal methods (Language Models-based methods) will be presented.

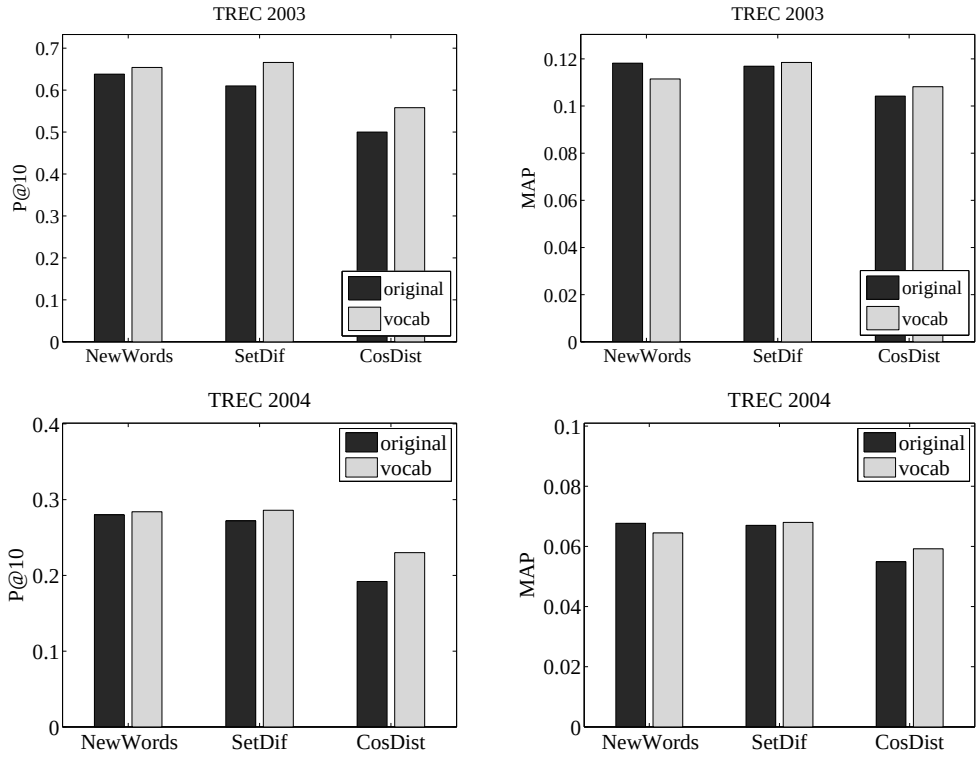


Figure 2.8: Performance of the standard methods and their variants using vocabulary pruning (vocabulary composed of all terms in top 25 retrieved sentences).

2.2.5 Language Modeling for the Novelty Task

A Statistical Language Model is a probabilistic mechanism for explaining the generation of text. It basically defines a distribution over all possible word sequences. The simplest LM is the unigram LM, which is a word distribution. In this work we employ unigram LMs, whose effectiveness for information retrieval tasks has been demonstrated in the literature [ZL01]. A simple LM for a document or sentence is the maximum likelihood estimator (*mle*), which associates a probability greater than zero for each term which appears in the document and a zero probability for the unseen terms. More specifically, for each term t, the probability $p_{mle}(t|s)$ represents the relative frequency of the term t in the sentence s. This estimator is problematic because assigning probabilities equal to zero to any unseen term is very strict. To overcome this problem, *mles* are often smoothed using some fallback model that suffers less from sparseness (e.g. a model constructed from a large collection of documents).

Smoothing techniques try to balance the probability of terms that appear in a document with those ones that are missing. It discounts the probability mass assigned to the seen words and distributes the extra probability to the unseen terms according to some fallback model.

Among the existing smoothing models, we will particularly utilize two of them: Jelinek-Mercer (JM) and Dirichlet (DIR) smoothing models, which have already been subject to study in the previous chapter for sentence retrieval purposes. Jelinek-Mercer smoothing involves a linear interpolation of the maximum likelihood model with the collection model, using a coefficient λ:

$$p(t|s) = (1-\lambda) \cdot p_{mle}(t|s) + \lambda \cdot p(t) = (1-\lambda) \cdot \frac{c(t,s)}{\sum_{t \in s} c(t,s)} + \lambda \cdot p(t) \quad (2.10)$$

where $p(t)$ is the *mle* constructed from the set of documents in the collection and $c(t,s)$ the term count of t in s.

Dirichlet smoothing adjusts the amount of reliance on the observed text according to the length of this text:

$$p(t|s) = \frac{c(t,s) + \mu \cdot p(t)}{\sum_{t \in s} c(t,s) + \mu} \quad (2.11)$$

where μ is the smoothing parameter. As argued in [Zha02], applying Dirichlet smoothing with query likelihood leads to a retrieval formula with components similar to the tf-idf weights and a document length correction.

In the literature, some novelty detection methods have been proposed based on Kullback-Leibler Divergence (KLD), which measures the divergence between two probability distributions. It can be used as a *distance* between LMs. KLD is always positive and greater than zero.

We apply here different methods to support novelty detection using the power and robustness of LMs. More specifically, we evaluate two main alternatives: Aggregate Model and Non-Aggregate Model. These models are described in the next subsections.

2.2.5.1 Aggregate vs. Non-Aggregate Models

The Aggregate Model (AM) and the Non-Aggregate Model (NAM) [AWB03] are two formal methods based on Language Modeling. Both of them utilize KLD to compute the novelty scores of sentences. KLD measures the divergence between two probability distributions (p_1 and p_2) as follows:

$$KLD(p_1||p_2) = \sum_x p(x|p_1) \cdot \log \frac{p(x|p_1)}{p(x|p_2)} \tag{2.12}$$

In the context of information retrieval, given two LMs associated to sentences s_i and s_j, the expression above can be rewritten as:

$$KLD(s_i||s_j) = \sum_t p(t|s_i) \cdot \log \frac{p(t|s_i)}{p(t|s_j)} \tag{2.13}$$

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

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

$$N_{NAM}(s_i|s_1, ..., s_{i-1}) = \min(KLD(s_i||s_1), ..., KLD(s_i||s_{i-1})) \tag{2.14}$$

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 considers the set of previous sentences as a whole. Therefore, it generates a LM for the current sentence and another LM for the set of previous sentences. This means that the contextual information is treated as a single unit that represents the user's inspection of the retrieved set of sentences. The novelty score is simply the KLD between the LM of the current sentence and the LM of the set of previously seen sentences, i.e.:

$$N_{AM}(s_i|s_1, ..., s_{i-1}) = KLD(s_i||s_1, ..., s_{i-1}) \tag{2.15}$$

In order to generate the LMs, we experimented with DIR and JM smoothing. With DIR smoothing we tested the following μ values: 1, 2, 10, 100, 500, 800, 1000, 2000, 3000, 4000, 5000 and 10000. In the case of JM smoothing the values assigned to λ were 0.01, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7 0.8 and 0.9.

Note that, in order to compute the KLD between a pair of LMs, we need to go over all terms in the vocabulary. With NAM, to obtain the novelty score for a sentence s_i we need to do $i-1$ KLD computations and, therefore, we need to traverse the whole vocabulary $i-1$ times. As we go down in the ranking of sentences, the number of iterations over the whole vocabulary increases and, consequently, the computational cost increases severely. In order to address this problem, in the next subsection we propose a simple,

effective and efficient modification of the KLD so that, instead of traversing all the terms of the vocabulary, we only go over a small subset of terms.

2.2.5.2 NAM-Quick: Efficient Non-Aggregate Model

NAM is computationally inefficient because it requires to go over all the vocabulary terms multiple times to compute the novelty score for each sentence. This is aggravated as we compute the novelty score for sentences in lower positions in the list because they require revisiting many previous sentences to estimate their novelty.

We propose here a method that alleviates this problem. This new technique, referred to as NAM-Quick, is a simple variation of NAM that computes an approximation of KLD. 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,

$$\text{KLD}^*(s_i||s_j) = \sum_{t \in s_i \vee s_j} P(t|s_i) \cdot \log \frac{P(t|s_i)}{P(t|s_j)} \tag{2.16}$$

We use the notation KLD^* to emphasize that this is an approximation to the real KLD value[6]. The novelty score is computed as the pairwise operations using this version of KLD:

$$N_{NAM\text{-}Quick}(s_i|s_1, ..., s_{i-1}) = \min(\text{KLD}^*(s_i||s_1), ..., \text{KLD}^*(s_i||s_{i-1})) \tag{2.17}$$

We expect that this is not only a more efficient method but also performs better 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 in 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.

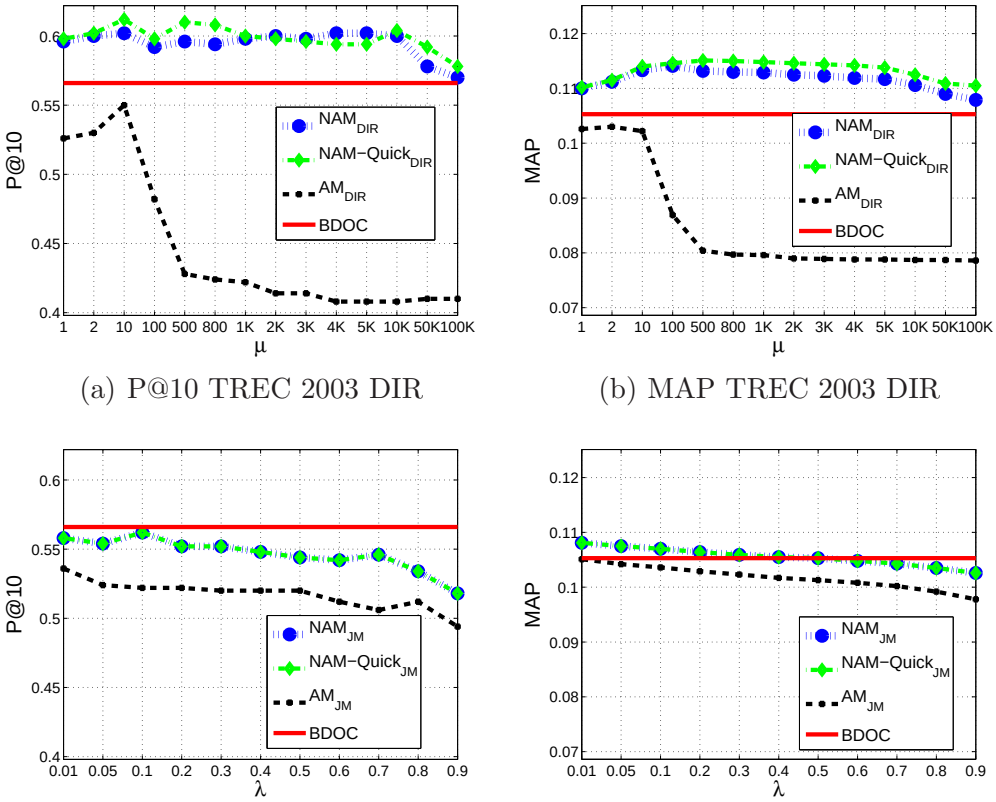
Table 2.6 compares the time performance of NAM and NAM-Quick with the mean time per query (in seconds)[7]. With NAM-Quick, time savings are substantial.

[6]We only study the performance of this variant with NAM because the problem of efficiency is especially serious with NAM.

[7]We executed the 50 queries in each collection on a quad-core machine (2.8 GHz) and computed the mean user+system time taken to process each query.

	NAM	NAM-Quick
TREC 2003	6.01	0.20
TREC 2004	13.96	0.40

Table 2.6: Average time (in seconds) needed to process a query with NAM and NAM-Quick.

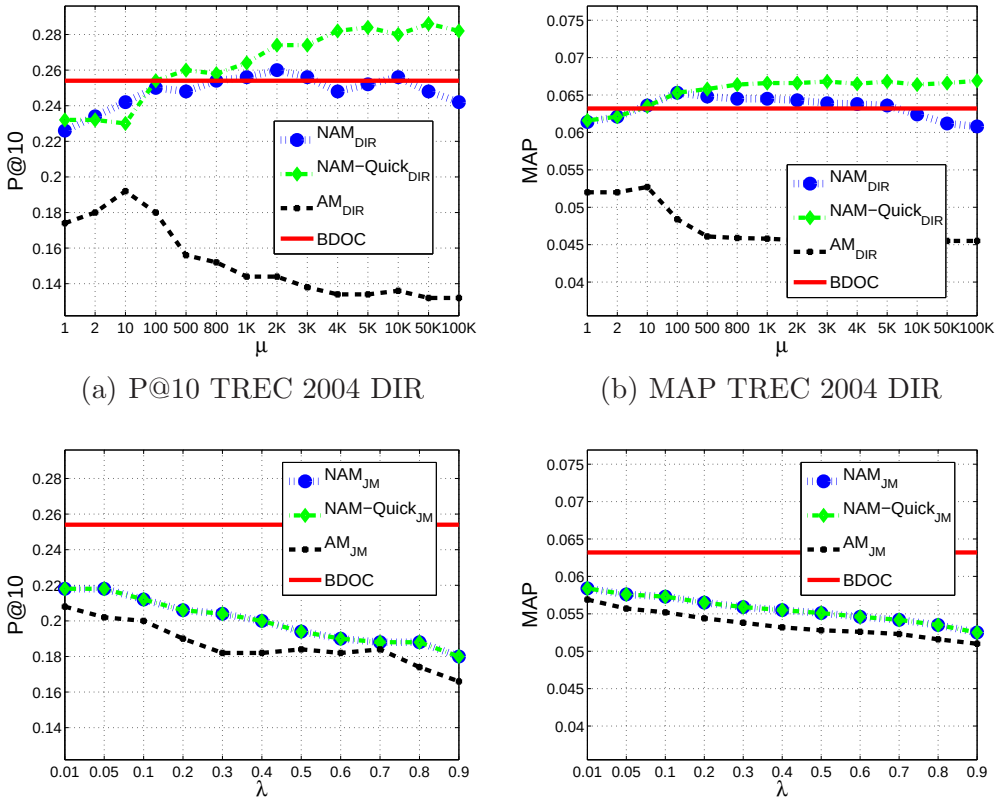


(a) P@10 TREC 2003 DIR (b) MAP TREC 2003 DIR

(c) P@10 TREC 2003 JM (d) MAP TREC 2003 JM

Figure 2.9: KLD-based models for novelty detection using TREC 2003 vs. BDOC baseline.

Figures 2.9 and 2.10 show the performance of the KLD-based models against the BDOC baseline given different values of the smoothing parameter. Aggregate Models (AM) perform worse than Non-Aggregate Models (NAM and NAM-Quick). Note that AM never outperforms the BDOC baseline (regardless of the smoothing method applied). NAM generates an individual LM for each sentence and, given a sentence s_i, its degree of novelty is estimated from the previously seen sentence having the smallest divergence. On the other hand, AM considers the history of seen sentences as a whole.



(a) P@10 TREC 2004 DIR (b) MAP TREC 2004 DIR

(c) P@10 TREC 2004 JM (d) MAP TREC 2004 JM

Figure 2.10: KLD-based models for novelty detection using TREC 2004 vs. BDOC baseline.

This seems to be harming. A LM for the set of seen sentences might be too general. Consider a sentence that is an exact repetition of a past sentence. With NAM, the sentence would receive the lowest novelty score. In contrast, with AM, this is not guaranteed. The larger the history is, the less important the terms of the sentence are in the LM of the seen sentences. Therefore, it is still possible that the sentence is classified as novel. Note also that AM performs worse as smoothing increases. This is quite natural because, as we make the LM more general (we give more importance to terms in the collection), terms seen in the history receive increasingly less importance.

Regarding the Non-Aggregate Models (NAM and NAM-Quick), the performance with DIR smoothing is better than the performance with JM. With JM smoothing, the proposed models hardly outperform the BDOC baseline (except for MAP in TREC 2003). JM is less sensitive than DIR to the tuning of the smoothing parameter. On the other hand, Non-Aggregate Models that

		DIR			JM		
	BDOC	**NAM**	**NAM-Quick**	**AM**	**NAM**	**NAM-Quick**	**AM**
		test: TREC 2003 (train: TREC 2004)					
P@10	.5660	**.6000**	.5920	.5500	.5580	.5580	.5360
$\Delta\%$		*(+6.01)*	*(+4.59)*	*(−2.83)*	*(−1.41)*	*(−1.41)*	*(−5.30)*
		($\mu = 2000$)	($\mu = 50000$)	($\mu = 10$)	($\lambda = 0.01$)	($\lambda = 0.01$)	($\lambda = 0.01$)
MAP	.1053	**.1141***	.1105	.1022	.1081	.1081	.1051
$\Delta\%$		*(+8.36)*	*(+4.94)*	*(−2.94)*	*(+2.66)*	*(+2.66)*	*(−0.19)*
		($\mu = 100$)	($\mu = 100000$)	($\mu = 10$)	($\lambda = 0.01$)	($\lambda = 0.01$)	($\lambda = 0.01$)
		test: TREC 2004 (train: TREC 2003)					
P@10	**.2540**	.2420	.2300	.1920	.2120	.2120	.2080*
$\Delta\%$		*(−4.72)*	*(−9.45)*	*(−24.41)*	*(−16.54)*	*(−16.54)*	*(−18.11)*
		($\mu = 10$)	($\mu = 10$)	($\mu = 10$)	($\lambda = 0.1$)	($\lambda = 0.1$)	($\lambda = 0.01$)
MAP	.0632	.0653	**.0658**	.0520*†	.0584	.0584	.0569*
$\Delta\%$		*(+3.32)*	*(+4.11)*	*(−17.72)*	*(−7.59)*	*(−7.59)*	*(−9.97)*
		($\mu = 100$)	($\mu = 500$)	($\mu = 2$)	($\lambda = 0.01$)	($\lambda = 0.01$)	($\lambda = 0.01$)

Table 2.7: KLD-based models evaluated in a training-testing setting.

apply DIR smoothing are generally able to outperform the BDOC baseline.

NAM-Quick performs the same as NAM with JM smoothing. Given the LMs for two sentences, s_i and s_j, both of them smoothed with JM mechanism, it can be demonstrated that the contribution of terms that do not belong to any of these sentences is null ($p(t|s_i) = p(t|s_j)$, for terms unseen in both sentences) and, therefore, $\text{KLD}(s_i||s_j) = \text{KLD}^*(s_i||s_j)$. However, with DIR, NAM-Quick leads to a different novelty ranking and performs slightly better than NAM.

To further check the models in a training-testing setting, we applied cross-validation to tune the smoothing parameters. We extracted the best configuration setting using one collection (training dataset) and this setting was used in the remaining collection (testing dataset). Table 2.7 and Figure 2.11 show the performance obtained and report the trained smoothing parameter settings. NAM's and NAM-Quick's performance is, in general, slightly higher than the performance of the baseline but we do not obtain many statistically significant differences with respect to BDOC. Observe also that NAM-Quick tends to perform better than NAM with most of the smoothing levels (Figures 2.9 and 2.10) but when it comes to a training-testing setting (Table 2.7) it does not perform better than NAM. Still, we would opt for NAM-Quick because of time efficiency reasons.

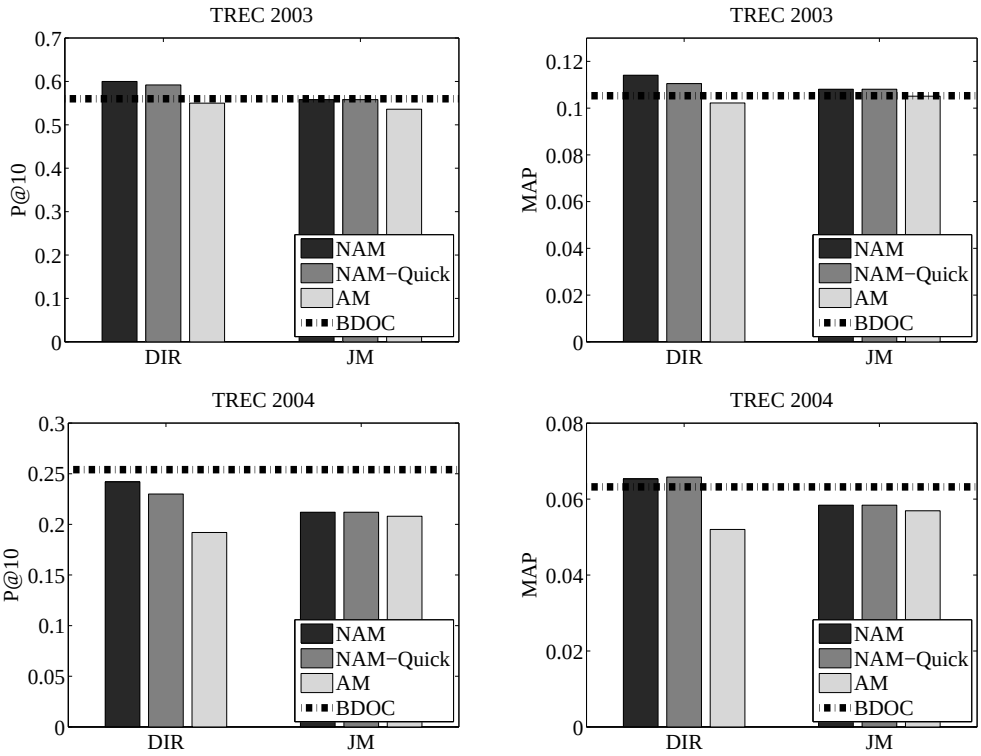


Figure 2.11: Comparison among the BDOC baseline and NAM, NAM-Quick and NAM by considering DIR and JM smoothing methods.

2.2.6 Mixture Model

One of the problems with AM and NAM is that they depend on the smoothing parameter and, therefore, we need to tune this parameter in order to smooth properly the LMs [Fer07]. In order to address this problem, there are mechanisms to adjust the internal parameters in an automatic way, i.e. to estimate automatically the parameters involved. A well-known method is the Expectation-Maximization (EM) algorithm [DLR77, MK08], which has been proved useful in different scenarios. For instance, in [ZCL03], 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 useful in the context of novelty detection at sentence level as defined in the TREC Novelty Tracks [Har02, SH03, Sob04]. We now describe the formulation of this model adapted to the detection of novel sentences and using the Expectation-Maximization algorithm.

The EM algorithm is a general method to find the maximum-likelihood estimate of the parameters of an underlying distribution from a given dataset

when the data is incomplete or has missing values. In the novelty detection scenario we use this algorithm in order to estimate these novelty scores. In Appendix C we make a careful presentation of the EM-algorithm while we focus here on its use for novelty detection.

We will consider that the set of samples $\mathcal{X}$ are documents (sentences), which are composed by N terms $\{x_1, ..., x_N\}$. This means that the length of a given document (sentence) is N.

We will only consider mixture models composed of two components: a background model (called θ_B) and a reference model (called θ_R). θ_R models a document, a sentence or a small set of documents or sentences, while θ_B models a large data set (e.g. the set of documents or sentences in a collection).

JM smoothing is a straightforward example of a mixture model. In this smoothing strategy, the maximum likelihood estimator (mle) of a document is interpolated with a background model (obtained, for instance, from a large data set). The JM expression for information retrieval is the following:

$$p(x_i|d) = (1 - \lambda) \cdot p_{mle}(x_i|d) + \lambda \cdot p(x_i) \tag{2.18}$$

If we consider that d is the reference model and the document/sentence collection is the background model we have:

$$p(x_i|\Theta) = (1 - \lambda) \cdot p(x_i|\theta_R) + \lambda \cdot p(x_i|\theta_B) \tag{2.19}$$

This can be interpreted as a Hidden Markov Model [MLS99]. Intuitively, given a query, each query term (sample) may have been extracted from a document the user has in mind or from a generic vocabulary which reflects the general usage of the language.

A Hidden Markov Model is composed by a set of states, a set of probabilities for the transitions between pairs of states, output symbols and a probability distribution on output symbols for each state. In our case, we have only two states: $state_R$, that is, "a term is extracted from the reference model (from an ideal document or sentence)" and $state_B$, that is, "a term is extracted from the background model (from the general knowledge the user has about the language)". Intuitively, the user traverses through these states when articulating a query. If the user query is a keyword query, then the user has mostly passed through $state_R$. In contrast, given a verbose query, the user may need to pass frequently through $state_B$ to generate lexical tissue.

The hidden variables (y_i) indicate which distribution generated each sample (term in the document or sentence). In other words, each $y_i \in \mathcal{Y}$ indicates whether the term $x_i \in \mathcal{X}$ was generated from the reference model θ_R or from the background model θ_B.

Since we are considering only two different models, there are only two weights associated to the probability distributions: α_1 and α_2. These α's measure the relative weight of θ_B and θ_R in the generation process. Considering JM (Equation 2.19), we have that $\alpha_1 = (1 - \lambda)$ and $\alpha_2 = \lambda$.

To sum up, our model is as follows:

- $\mathcal{X} = \{x_1, ..., x_n\}$ is a document or sentence composed by N terms.
- $\Theta = \{\theta_R, \theta_B\}$.
- $y = \{y_1, ...y_N\}$ (where $y_i = \{1, 2\}$, $i = 1, ..., N$) indicates which distribution generated each term ($y_i = 1$ means that x_i was generated from θ_R and $y_i = 2$ means that x_i was generated from θ_B).
- There is only one parameter to learn: λ. The factors for the two components of the mixture model are:

  (i) $\alpha_1 = (1 - \lambda)$, for θ_R

  (ii) $\alpha_2 = \lambda$, for θ_B

The expressions for each step of the EM algorithm are:

(i) **E-Step**:

$$p(y_i = 1|x_j, \Theta) = \frac{\alpha_1 \cdot p(x_j|\theta_R)}{\alpha_1 \cdot p(x_j|\theta_R) + \alpha_2 \cdot p(t_j|\theta_B)} \tag{2.20}$$

$$p(y_i = 2|x_j, \Theta) = \frac{\alpha_2 \cdot p(x_j|\theta_B)}{\alpha_1 \cdot p(x_j|\theta_R) + \alpha_2 \cdot p(t_j|\theta_B)} \tag{2.21}$$

(ii) **M-Step**:

$$\alpha_1 = \frac{1}{N} \cdot \sum_{j=1}^{N} p(y_i = 1|x_j, \Theta) \tag{2.22}$$

$$\alpha_2 = \frac{1}{N} \cdot \sum_{j=1}^{N} p/y_i = 2|x_j, \Theta) \tag{2.23}$$

This iterative process allows us to learn automatically the weights associated to this two-mixture model. Given a sample (document or sentence) we would only need to initialize randomly the α_i's and, next, run the EM algorithm to obtain an estimation of λ. Now, we describe how to use EM in our scenario.

2.2.6.1 The EM Algorithm and Novelty Detection

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 [ZCL03], 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 [ZCL03]:

$$l(\lambda|s_i, \theta_{R_s}) = \sum_{j=1}^{c(s_i)} \log((1-\lambda) \cdot p(t_j|\theta_{R_s}) + \lambda \cdot p(t_j|\theta_B)) \tag{2.24}$$

where s_i is the current sentence, $c(s_i)$ is the number of terms in sentence s_i, θ_{R_s} is the reference model (it models the history of sentences), and θ_B is the background model (it models the use of terms in a large collection).

The parameter λ indicates if a term is more probable to have been extracted from θ_B or θ_R. We can assume that the final value of λ is the novelty score for a specific sentence: if a sentence is more probable to have been extracted from θ_R than from θ_B then this sentence is more salient in θ_R than in θ_B. So, the sentence is hardly novel because it has more weight in the seen sentences than in the general collection. On the other hand, when a sentence is more probable to have been extracted from the θ_B model (the model of the collection) than from θ_R, it indicates that it might be a novel sentence.

So, given a sentence s_i, we want to estimate its novelty score through the estimation of λ. Formally:

$$l(\lambda|s_i, \theta_R) = \sum_{j=1}^{c(s_i)} \log\big((1-\lambda) \cdot p(x_j|\theta_R) + \lambda \cdot p(x_j|\theta_B)\big) \tag{2.25}$$

If we assume that θ_R is a model for the set of previously seen sentences ($\{s_1, ..., s_{i-1}\}$), the novelty score for the sentence s_i is:

$$N(s_i; \theta_R) = arg\ max_\lambda l(\lambda|s_i, \theta_R) \tag{2.26}$$

We can apply the EM algorithm to find the value of λ that maximizes this score. In Figure 2.12 we show a pseudocode that implements the estimation of λ given a sentence, Θ_R and Θ_B.

If we model individually each previous sentence then we have $i-1$ reference models: $\theta_{R_{s_1}}, \theta_{R_{s_2}}, \ldots, \theta_{R_{s_{i-1}}}$. In this case, the novelty score associated

EM-algorithm

Input: θ_R, θ_B, *sentence* ($s_i = \{t_1, t_2, ..., t_n\}$) .
Output: λ

$\alpha_1 = 0.5$
$\alpha_2 = 0.5$
$prev = MIN\_SCORE$

do {
// E-Step
for each $t_j \in s_i$ {
compute $p(y_i = 1|t_j \in s_i, \Theta) = \frac{\alpha_1 \cdot p(t_j|\theta_R)}{\alpha_1 \cdot p(t_j|\theta_R) + \alpha_2 \cdot p(t_j|\theta_B)}$
compute $p(y_i = 2|t_j \in s_i, \Theta) = \frac{\alpha_2 \cdot p(t_j|\theta_B)}{\alpha_1 \cdot p(t_j|\theta_R) + \alpha_2 \cdot p(t_j|\theta_B)}$
}
// M-Step
compute $\alpha_1 = \frac{1}{n} \sum_{j=1}^{n} p(y_i = 1|t_j \in s, \Theta)$
compute $\alpha_2 = \frac{1}{n} \sum_{j=1}^{n} p(y_i = 2|t_j \in s, \Theta)$

compute $l(\alpha_1, \alpha_2|s_i, \theta_R) = \sum_{j=1}^{n} \log\left(\alpha_1 \cdot p(t_j|\theta_R) + \alpha_2 \cdot p(t_j|\theta_B)\right)$

// stop condition (continues iterating while $l(\alpha_1, \alpha_2|s_i, \theta_R)$ decreases)
if ($l(\alpha_1, \alpha_2|s_i, \theta_R) <= prev$)
return α_2
$prev = l(\alpha_1, \alpha_2|s_i, \theta_R)$

} while ($true$)

Figure 2.12: Application of the EM algorithm to estimate λ for novelty detection purposes.

to sentence s_i can be the minimum or the average of the $i-1$ computed scores:

$$N_{MMEM\text{-}NAM_{min}}(s_i|s_1, ..., s_{i-1}) = \min(N_{score}(s_i; \theta_{R_{s_1}}), ..., N_{score}(s_i; \theta_{R_{s_{i-1}}})) \tag{2.27}$$

$$N_{MMEM\text{-}NAM_{avg}}(s_i|s_1, ..., s_{i-1}) = \text{avg}(N_{score}(s_i; \theta_{R_{s_1}}), ..., N_{score}(s_i; \theta_{R_{s_{i-1}}})) \tag{2.28}$$

Observe that this is similar to a Non-Aggregate approach (NAM) in KLD-based models (sentences in the history modeled individually). We therefore

NOVELTY (MMEM-AM)

Input: an ordered set of sentences $\{s_1, ..., s_n\}$
Output: novelty scores for each sentence in the initial ranking (N_{score})

$N_{score}(s_1) = MAX\_SCORE$

build θ_B given the collection

for $i = 2$ to n {
    build θ_R as the *mle* of $s_1, ..., s_{i-1}$
    $N_{score}(s_i|s_1, ..., s_{i-1}) = \text{EM-algorithm}(\theta_R, \theta_B, s_i)$
}

Figure 2.13: Pseudocode for novelty detection with the EM algorithm and an Aggregated approach.

refer to these models as MMEM-NAM$_{min}$ and MMEM-NAM$_{avg}$, respectively. In contrast, an AM-like approach (labeled as MMEM-AM) that uses the EM-algorithm, given a reference model $\theta_{R_{s_1,...,s_{i-1}}}$ and a background model θ_B, is defined as:

$$N_{MMEM\text{-}AM}(s_1|s_1, ..., s_{i-1}) = N_{score}(s_i; \theta_{R_{s_1,...,s_{i-1}}}) \tag{2.29}$$

The pseudocode for computing the novelty scores given the ranking of sentences is presented in Figure 2.13 (MMEM-AM) and Figure 2.14 (MMEM-NAM$_{min}$ and MMEM-NAM$_{avg}$).

Note that, in order to generate θ_R, we need to use a mechanism of smoothing such as JM or DIR. The smoothing parameter values are reported in Table 2.8 (these values were fixed with a train-test approach). Table 2.9 and Figure 2.15 report the performance obtained with these methods. Among the approaches proposed here, MMEM-NAM$_{min}$ is the variant that performs the best (independently of the smoothing mechanism). Usually, with MMEM-NAM$_{min}$, DIR smoothing is the most appropriate smoothing method. Note that, although this approach performs better than the BDOC baseline (except for P@10 in TREC 2004), in most of cases no statistical significant differences are attained.

Instead of introducing smoothing for θ_R, we could have used the maximum likelihood estimator. However, we demonstrated in [FL08] that this non-smoothed model is less competitive.

To sum up, although MMEM-NAM$_{min}$ is able to outperform the baseline in terms of P@10 (TREC 2004) and MAP (TREC 2003 and 2004), usually no

NOVELTY (MMEM-NAM)

Input: an ordered set of sentences $\{s_1, ..., s_n\}$
Output: novelty scores for each sentence in the initial ranking

$N_{score}(s_1) = MAX\_SCORE$

build θ_B given the collection

for $i = 2$ to n {
    create an empty table of $i-1$ scores (NS)
    for each sentence $s_j \in \{s_1, ..., s_{i-1}\}$ {
        build $\theta_{R_{s_j}}$ as the mle given s_j
        $NS(s_j) =$ EM-algorithm$(\theta_{R_{s_j}}, \theta_B, s_i)$
    }
    $N_{score}(s_i|s_1, ..., s_{i-1}) = \min/\text{avg}(NS)$
}

Figure 2.14: Pseudocode for novelty detection with the EM algorithm and a Non-Aggregate approach.

	DIR			**JM**		
	MMEM-AM	**MMEM-NAM$_{min}$**	**MMEM-NAM$_{avg}$**	**MMEM-AM**	**MMEM-NAM$_{min}$**	**MMEM-NAM$_{avg}$**
	test: TREC 2003 (train: TREC 2004)					
P@10	μ=10000	μ=50	μ=25	λ=0.1	λ=0.1	λ=0.1
MAP	μ=100000	μ=25	μ=1	λ=0.1	λ=0.1	λ=0.1
	test: TREC 2004 (train: TREC 2003)					
P@10	μ=50000	μ=5	μ=1	λ=0.1	λ=0.4	λ=0.1
MAP	μ=1	μ=5	μ=1	λ=0.1	λ=0.1	λ=0.1

Table 2.8: Smoothing parameter values (μ/λ) for DIR and JM when building θ_R.

statistical significant results are obtained. Therefore, our MMEM approaches are not effective enough for novelty detection.

2.2.7 Comparing Novelty Methods

Along this section we have proposed and tested different novelty detection approaches. For each technique, we explained and motivated the methods (some of them are variants of existing techniques and other methods are novel definitions within formal frameworks). Now, we analyze the best per-

		DIR			JM		
	BDOC	**MMEM-AM**	**MMEM-NAM$_{min}$**	**MMEM-NAM$_{avg}$**	**MMEM-AM**	**MMEM-NAM$_{min}$**	**MMEM-NAM$_{avg}$**
	test: TREC 2003 (train: TREC 2004)						
P@10	.5660	.5520	.6040	.4980*	.5600	**.6060**	.5420
$\Delta\%$		(-2.47)	$(+6.71)$	(-12.01)	(-1.06)	$(+7.07)$	(-4.24)
MAP	.1053	.1078	**.1130***	.1021	.1096	.1129	.1022
$\Delta\%$		$(+2.37)$	$(+7.31)$	(-3.04)	$(+4.08)$	$(+7.22)$	(-2.94)
	test: TREC 2004 (train: TREC 2003)						
P@10	.2540	**.2560**	.2480	.2060	.2360	.2480	.2120
$\Delta\%$		$(+0.79)$	(-2.36)	(-18.90)	(-7.09)	(-2.36)	(-16.54)
MAP	.0632	.0611	**.0633**	.0582	.0617	.0621	.0582
$\Delta\%$		(-3.32)	$(+0.16)$	(-7.91)	(-2.37)	(-1.74)	(-7.91)

Table 2.9: Performance of the MMEM novelty detection methods considering the BDOC baseline.

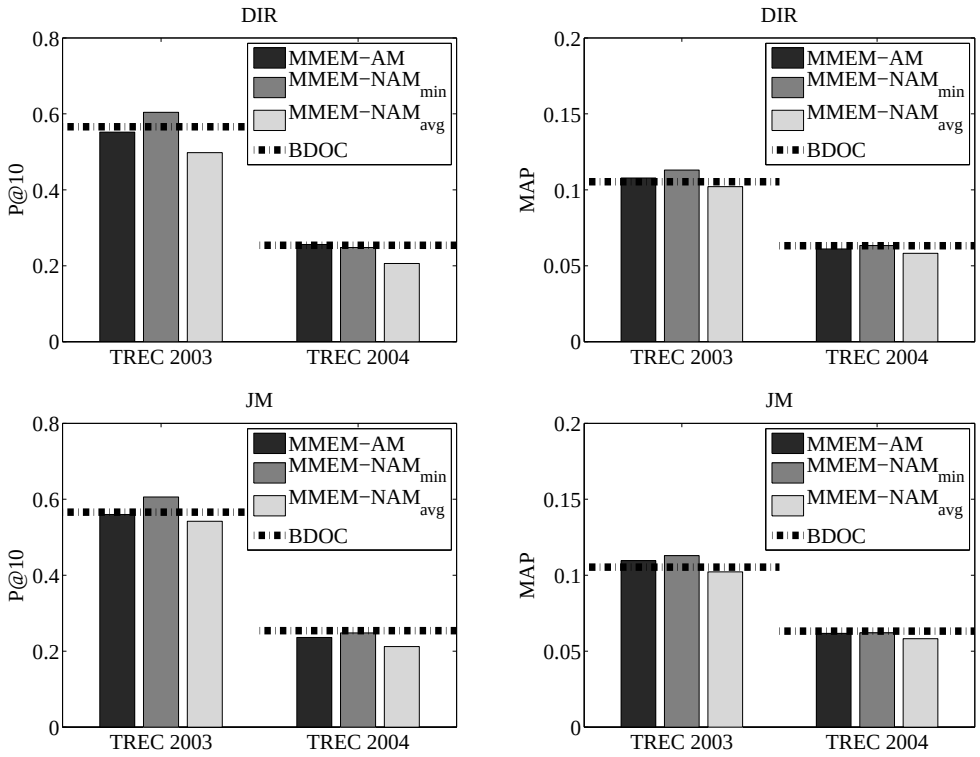


Figure 2.15: Comparison between mixture model approaches and the BDOC baseline.

forming models: NewWords, SetDif with vocabulary pruning, NAM-Quick and MMEM-NAM$_{min}$, and compare them against the BDOC baseline.

In Table 2.10 and Figure 2.16 we compare the performance of different

	BDOC	NewWords	SetDif (vocab.)	NAM-Quick (DIR)	MMEM-NAM$_{min}$
			test: TREC 2003 (train: TREC 2004)		
P@10	.5600	.6380*†	**.6660***†	.5920	.6040
Δ%		(+12.72)	(+18.93)	(+4.59)	(+6.71)
MAP	.1053	.1182	**.1185***†	.1105	.1130*
Δ%		(+12.25)	(+12.54)	(+4.94)	(+7.31)
			test: TREC 2004 (train: TREC 2003)		
P@10	.2540	.2800	**.2860**	.2300	.2480
Δ%		(+10.24)	(+12.60)	(−9.45)	(−2.36)
MAP	.0632	.0677	**.0680**	.0658	.0633
Δ%		(+7.12)	(+7.59)	(+4.11)	(+0.16)

Table 2.10: Comparison of different novelty detection approaches against the BDOC baseline.

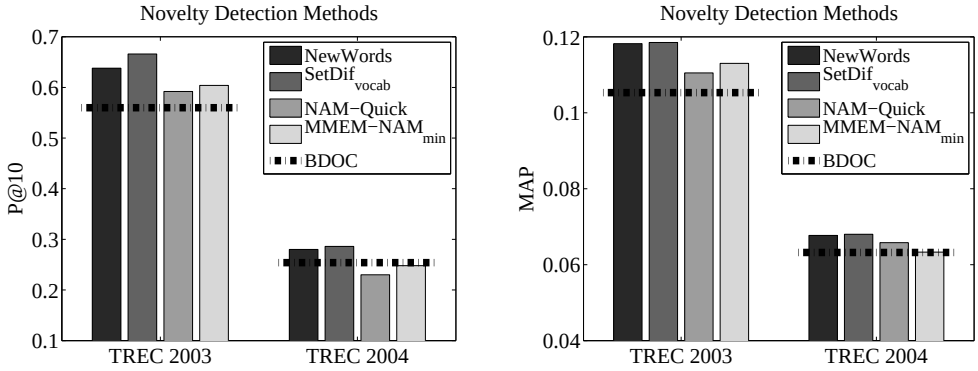


Figure 2.16: Comparison between different novelty detection methods and the BDOC baseline.

novelty detection methods against the baseline. Note that NAM-Quick and MMEM-NAM$_{min}$ do not outperform the baseline in terms of P@10 in TREC 2004. Furthermore, MMEM-NAM$_{min}$ only performs better than NAM-Quick in TREC 2003. NewWords and SetDif with vocabulary pruning perform better than NAM-Quick and MMEM-NAM$_{min}$. Note that NewWords only improves statistically significant the baseline in terms of P@10 in TREC 2003, while SetDif with vocabulary pruning leads to statistically significant improvements in both metrics in TREC 2003. However, in TREC 2004, no improvements are statistically significant.

To sum up, simple ad-hoc methods such as NewWords or SetDif are able to outperform the BDOC baseline. Furthermore, performance increases after pruning the vocabulary. SetDif is the method that performs the best when

a vocabulary obtained from the top 25 sentences of the BDOC ranking is considered. On the other hand, methods based on formal techniques, such as Language Models, are able to outperform the BDOC baseline in some cases but none of these methods are able to perform statistically better than the baseline in both collections.

We have modestly improved the performance of BDOC and some state of the art novelty detection measures by incorporating a vocabulary pruning into standard methods. However, the results obtained so far are not satisfactory. In the next section we evaluate the novelty detection methods proposed in a perfect relevance scenario and, next, we propose further improvements that handle the novelty detection problem more effectively.

2.3 Perfect Relevance

In Section 2.2 we reported experiments using a ranked set of estimated relevant sentences (non-perfect relevance). As argued in the beginning of this chapter, this non-perfect relevance evaluation is realistic. However, another alternative is to study novelty in a perfect relevance scenario. This helps to understand the relative merits of novelty methods without interferences coming from non-relevant material.

We made a complete pool of experiments evaluating the same strategies presented in Section 2.2. The only difference is that novelty detection techniques start now from the sentences judged as relevant by the TREC assessors and we follow the same order followed by the judges when estimating novelty. This corresponds with Task 2 (given the relevant sentences in all documents, identify all novel sentences) in the TREC Novelty Tracks.

For simplicity, we summarize now the main findings extracted from these experiments (making special emphasis on the differences found with respect to the non-perfect relevance experiments). The complete set of tables and figures are reported in Appendix D.

The main results are summarized in Table 2.11 and Figure 2.17. In terms of P@10, NewWords with vocabulary pruning is the only method that outperforms the baseline but the improvements are not statistically significant. In terms of MAP, NewWords$_n$ is the method that outperforms the baseline in both collections. However, note that only in TREC 2003 the improvement is statistically significant.

Summing up, with perfect relevance, the improvements over the baseline are weaker and less consistent than the improvements found with non-perfect relevance. We therefore investigate in the next section how to further refine novelty detection.

	Baseline	NewWords	NewWords$_n$	NewWords (vocab.)	NAM-Quick (DIR)	MMEM-NAM$_{min}$
			test: TREC 2003 (train: TREC 2004)			
P@10	.8760	.8800	.8360	**.9060**	.8340	.8500
Δ%		(+0.46)	(−4.57)	(+3.42)	(−4.79)	(−2.97)
MAP	.7411	**.8188***†	.8121*†	.7169	.8087*†	.8120*†
Δ%		(+10.48)	(+9.58)	(−3.27)	(+9.12)	(+9.57)
			test: TREC 2004 (train: TREC 2003)			
P@10	.7640	.6760	.6960	**.7740**	.6720*	.6820*
Δ%		(−11.52)	(−8.90)	(+1.31)	(−12.04)	(−10.73)
MAP	**.6103**	.6086	.6166	.5696	.6054	.5996
Δ%		(−0.28)	(+1.03)	(−6.67)	(−0.80)	(−1.75)

Table 2.11: Comparison of different novelty detection approaches against the perfect relevance baseline.

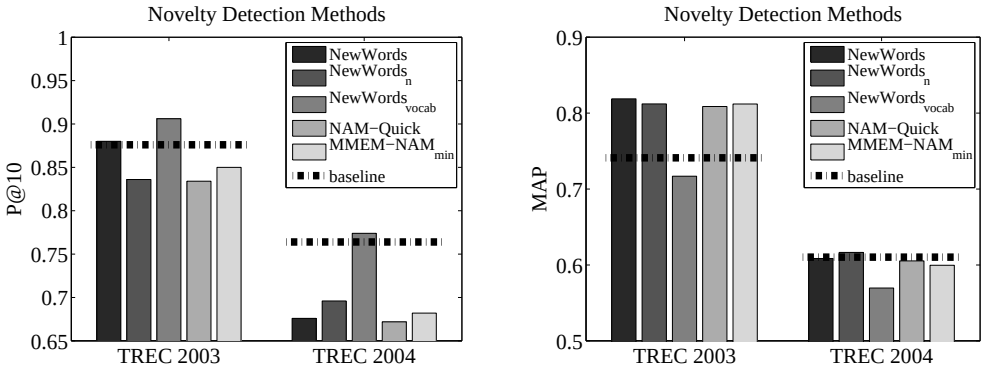


Figure 2.17: Comparison between different novelty detection methods and the perfect relevance baseline.

2.4 Novelty Detection Applied from a Given Position

In previous sections we presented novelty detection methods that provide modest improvements over BDOC and perfect relevance baselines. Our intuition is that, given a ranked set of sentences, the novelty detection measures proposed so far lead to a strong re-ordering of the sentences given a novelty criterion. However, the performance results indicate that such re-ordering does not help to obtain better effectiveness, especially in a perfect relevance scenario.

In this section we start from the perfect relevance ranking as the input of novelty detection methods because this baseline is harder to beat than the non-perfect relevance baseline and, therefore, this represents a major

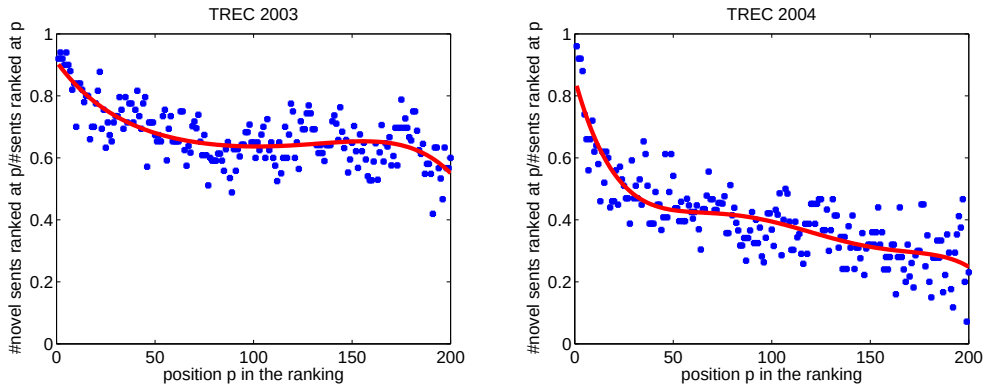


Figure 2.18: Proportion of novel sentences against rank (perfect relevance case).

challenge for novelty detection models.

The methods proposed along this section compute novelty starting from early positions in the rank. Nevertheless, filtering out sentences from top positions may be harmful because, usually, top-ranked sentences tend to be novel. In order to further support this claim, we show in Figure 2.18 the proportion of sentences, located at a position p, that are novel given the total amount of relevant sentences placed at such position p. We considered the 50 queries from TREC 2003 and 2004[8]. Given the top positions (e.g. $p < 100$), the proportion of novel sentences in the relevance ranking is very high. Although this is not surprising (initially, the user knows nothing and, therefore, everything is novel), our intuition is that the novelty methods described above (e.g. NewWords) lead to a strong re-ordering of the initial rank where sentences can be severely demoted in the ranking. This is likely harmful. To further illustrate this, we compare the percentage of novel sentences given the perfect relevance and NewWords rankings in Figure 2.19 (for the top 30 sentences).

Note that, at top ranked positions, NewWords does not retrieve as many novel sentences as the baseline. Therefore, re-ordering these initial sentences harms performance.

Given a novelty detection method, we propose here to preserve the original order for sentences located in the top $p-1$ positions. The remaining sentences, from position p to the end of the ranking of relevant sentences for the query, are re-ordered following the novelty metric. Note that top-sentences are *frozen* in the ranking but they are taken into account to estimate the novelty scores of the remaining sentences (they conform the history

[8]The line represents an approximation for the cloud of points using the least squares method.

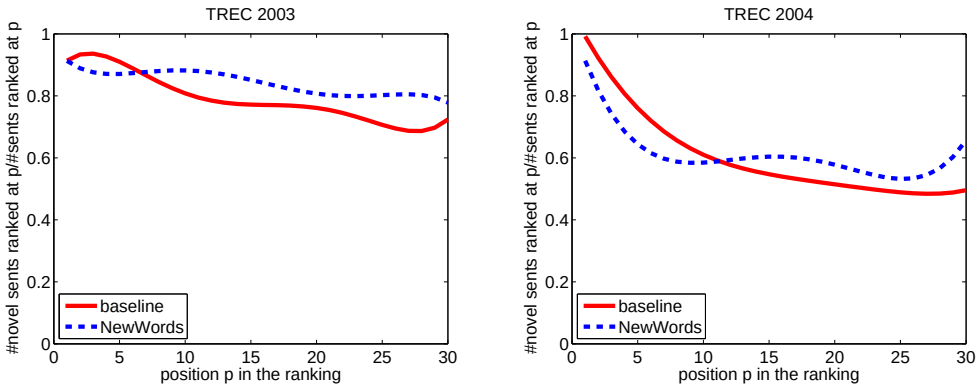


Figure 2.19: Proportion of novel sentences against rank (perfect relevance and NewWords case).

of seen material). The main challenge is to determine position p. To this aim, we propose two variants: a) a query-independent position, which consists of fixing the same value p for all queries, and b) a query-dependent position, which depends on the amount of redundancy in the ranking for the query, such that different queries may have different values of p. In both cases, state of the art novelty detection methods, i.e. NewWords, SetDif and CosDist, are considered to detect novelty from position p and the effectiveness of the variants proposed here is compared against the performance of the original state of the art novelty detection methods.

2.4.1 Novelty Detection with a Query-Independent Threshold

Our preliminary analysis suggested that novelty detection should not be applied from early positions. Therefore, we need to determine the position where we should start the novelty detection process. We propose first a simple method that consists of fixing such position (p) by applying a train-test mechanism: we train the value of p (the position that, applying novelty from such position, provides us with the highest performance in the training collection, TREC 2003) and, next, we use it as the position where we start detecting novelty in TREC 2004. The alternative train-test configuration (train with TREC 2004 and test with TREC 2003) was also tested. Given p and a sentence s_i ranked initially at position i, the novelty score of such sentence is computed as:

$$N_{meth_{q\text{-}i}}(s_i|s_1,...,s_{i-1}) = \begin{cases} \max_{val} - i & \text{, if } i < p \\ N_{meth}(s_i|s_1,...,s_{i-1}) & \text{, otherwise} \end{cases} \quad (2.30)$$

	NewWords	**NewWords$_{qi}$**	**SetDif**	**SetDif$_{qi}$**	**CosDist**	**CosDist$_{qi}$**
			test: TREC 2003 (train: TREC 2004)			
P@10	.8800	**.8900**	.8460	.8760	.8060	.8760*†
		(p=9)		(p=10)		(p=10)
Δ%		(*+1.14*)		(*+3.55*)		(*+8.68*)
MAP	.8188	**8202**	.7902	.7836	.8046	.8066
		(p=9)		(p=51)		(p=17)
Δ%		(*+0.17*)		(*−0.84*)		(*+0.25*)
		test: TREC 2004 (train: TREC 2003)				
P@10	.6760	**.7560***†	.5900	.7340*†	.6460	.7340*†
		(p=7)		(p=8)		(p=9)
Δ%		(*+11.83*)		(*+24.41*)		(*+13.62*)
MAP	.6086	**.6350***†	.5574	.6078*†	.5865	.6152*†
		(p=6)		(p=9)		(p=6)
Δ%		(*+4.34*)		(*+9.04*)		(*+4.89*)

Table 2.12: Comparison of performance between state of the art novelty detection methods and their variants based on a query-independent threshold.

where $\max_{val}$ is the highest possible novelty score (used to preserve the order of the top $p-1$ sentences) and $meth$ is one of the novelty detection methods: NewWords, SetDif or CosDist.

In our experiments, we tested values for p from 0 to 100, in steps of 1. In Table 2.12 we report the performance of the variants proposed here (labeled as NewWords$_{qi}$, SetDif$_{qi}$ and CosDist$_{qi}$) against the corresponding standard methods. We also indicate the trained positions (p) used in the test stage.

The new variant yields usually to statistically significant improvements over the original method. NewWords$_{qi}$ is the approach that performs the best and SetDif is the novelty approach that obtains the highest improvements after incorporating the threshold (up to 24%). Observe that, given the p values learned, the novelty process should still start at early positions.

These outcomes support our intuition and demonstrate that top-ranked sentences are novel and should not be re-ordered at the time of computing novelty.

In Figure 2.20 we show graphically the performance of the threshold-based variant, the original novelty detection methods and the baseline (perfect relevance with no novelty). In terms of P@10, the new variant outperforms the original novelty detection methods, but it does not outperform the baseline. However, in terms of MAP, this variant outperforms (or, at least, performs similarly to) the baseline. This is an important outcome that we had not obtained so far. Thus, these improvements in MAP support our initial hypothesis about the interest of freezing the initial ranking.

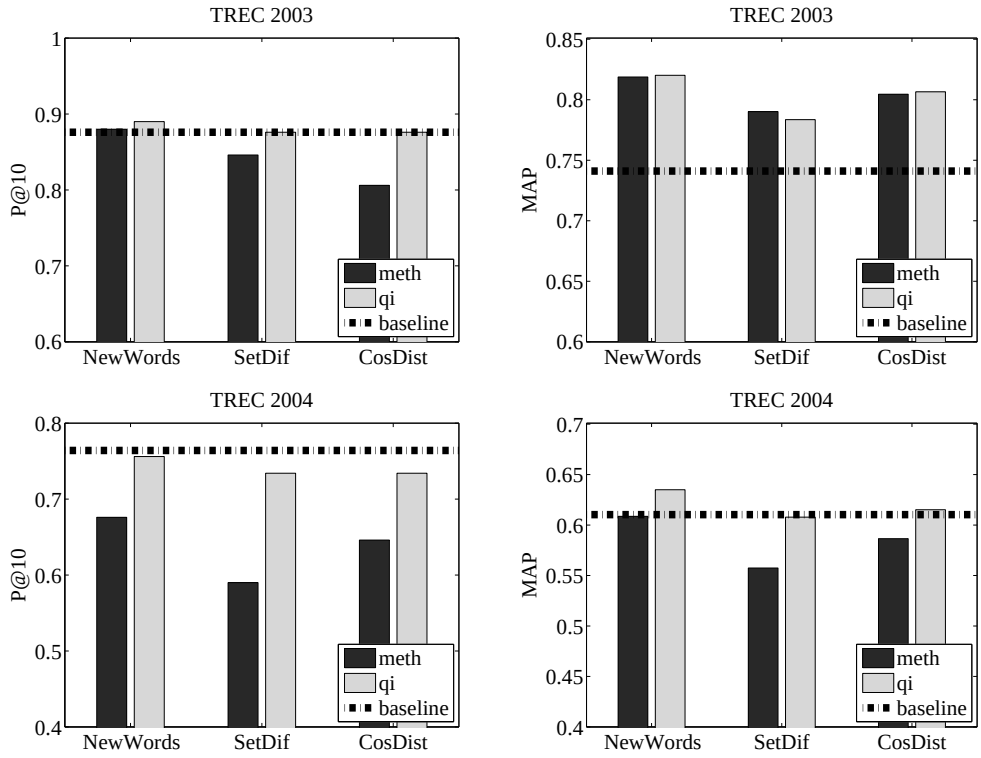


Figure 2.20: Comparison among the variant proposed here, the corresponding original versions and the perfect relevance baseline.

2.4.2 Novelty Detection with a Query-Dependent Threshold

In the previous subsection we applied a simple train-test method to define the position where we should start detecting novelty. The novelty process starts always at the same position for all queries. Nevertheless, this may not be the best approach because, depending on the query, we might find redundancy earlier or later in the rank. To address this problem, we propose here a query-dependent mechanism consisting of, for each query, estimating the position where we should start the novelty detection process. Two different variants are proposed: a cluster-based approach and a normalized-score approach.

2.4.2.1 Cluster-Based Approach

In this subsection we propose a method that drives novelty detection so that, depending on the query, the novelty detection process is triggered starting at a given position p in the ranking of sentences (the sentences in previous positions preserve their order). To determine the value of p we propose a

	NW	**NW**$_{qi}$	**NW**$_{qd_c}$	**SD**	**SD**$_{qi}$	**SD**$_{qd_c}$	**CD**	**CD**$_{qi}$	**CD**$_{qd_c}$
	test: TREC 2003 (train: TREC 2004)								
P@10	.8800	.8900	.8960	.8460	.8760	**.8980***	.8060	.8760*†	.8920*†
			(t=0.55)			(t=0.55)			(t=0.55)
Δ%		(+1.14)	(+1.82)		(+3.55)	(+6.15)		(+8.68)	(+10.67)
MAP	.8188	.8202	**.8237**	.7902	.7836	.7985	.8046	.8066	.8078
			(t=0.40)			(t=0.95)			(t=0.90)
Δ%		(+0.17)	(+0.60)		(−0.84)	(+1.05)		(+0.25)	(+0.40)
	test: TREC 2004 (train: TREC 2003)								
P@10	.6760	.7560*†	**.7800***†	.5900	.7340*†	.7700*†	.6460	.7340*†	**.7800***†
			(t=0.30)			(t=0.70)			(t=0.75)
Δ%		(+11.83)	(+15.38)		(+24.41)	(+30.51)		(+13.62)	(+20.74)
MAP	.6086	.6350*†	**.6389***†	.5574	.6078*†	.6169*†	.5865	.6152*†	.6340*†
			(t=0.50)			(t=0.50)			(t=0.55)
Δ%		(+4.34)	(+4.98)		(+9.04)	(+10.67)		(+4.89)	(+8.10)

Table 2.13: Comparing query-dependent thresholding (cluster-based), query-independent thresholding and no-thresholding (original method).

cluster-based approach. The intuition behind this idea is that novelty detection should only be started when we find some evidence about redundancy, i.e. a sentence is strongly thematically related to a previous one, and this can be detected using clustering. The k-NN clustering algorithm was widely used for cluster-based document retrieval, see [LCA08] for instance. Here we use a variant of the k-NN algorithm: instead of setting the number k of neighbors for a sentence we set the minimum similarity threshold t for the given metric (in our case cosine distance).

Given a sentence s_i, its *neighborhood* is the set of sentences s_k such that $sim(s_i, s_k) \geq t$. The method works as follows: first, we cluster all sentences in the collection using t-NN. Next, we scan sequentially the ranking of sentences and fix p to the position of the first sentence whose cluster (neighborhood) contains a sentence already seen before. This means that positions from 1 to $p-1$ are frozen, while sentences starting at the p position are re-ranked using the novelty detection methods described above.

In Table 2.13[9] and Figure 2.21 we compare the performance of the cluster-based approach, the query-independent approach and the standard novelty detection methods[10]. Independently of the novelty detection method applied, the approach proposed here improves the original novelty detection methods. Most of the improvements are statistically significant.

Note that the new variant, applied to all these methods, not only outperforms their corresponding original novelty method but also outperforms the

[9]We abbreviated NewWords, SetDif and CosDist with NW, SD and CD, respectively.
[10]Unlike in [FPLB10], we use here the same train-test methodology used so far.

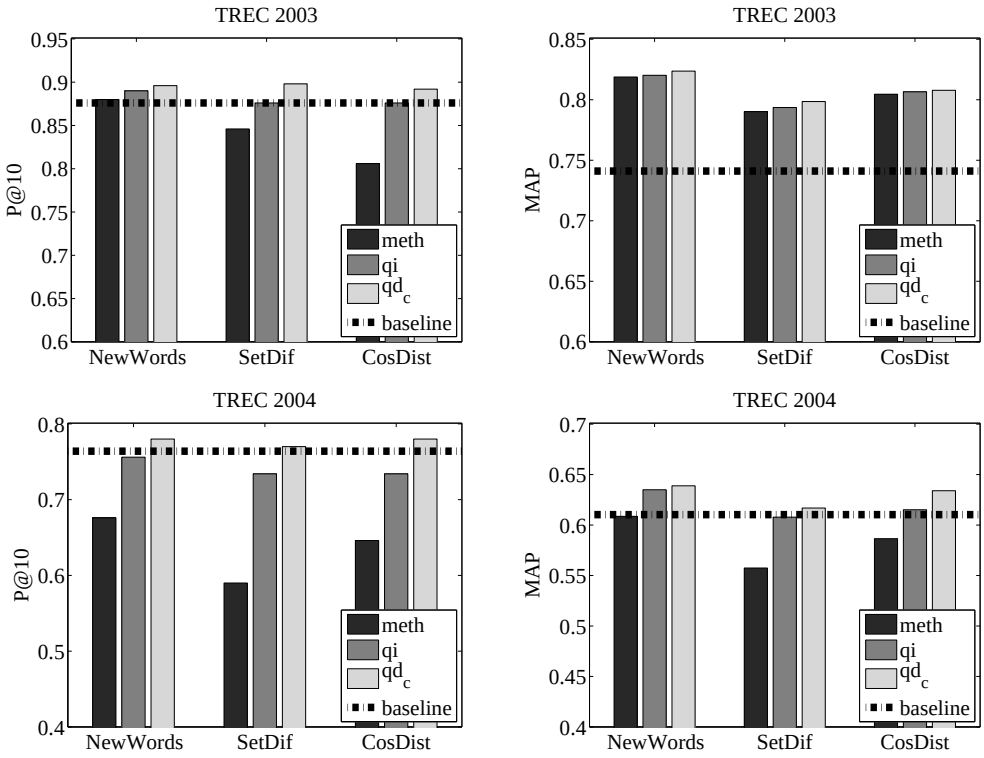


Figure 2.21: Comparing query-dependent thresholding (cluster-based), query-independent thresholding and no-thresholding (original method) against the perfect relevance baseline.

perfect relevance baseline, mostly in terms of MAP. These results indicate that this approach is effective. Moreover, although it is necessary a preprocessing step consisting of creating clusters of sentences, the clustering process can be computed at indexing time and, therefore, little computation cost is introduced at query time.

2.4.2.2 Normalized-Score Approach

In this subsection we propose another alternative that does not require clustering and which is based on a normalized score threshold. First, given a novelty detection method *meth* and ignoring the first sentence (which is always novel), we compute the novelty scores for all the relevant sentences for a query q. Given these novelty scores, we obtain the maximum novelty score max_q and, next, we normalize the computed scores for all sentences by dividing by max_q. We repeat this process for all queries. As a result of such normalization we obtain novelty scores within the range [0,1]. Given a

threshold $ns \in [0, 1]$, which indicates the level of novelty of a sentence, and the original relevance ranking, we start to compute novelty from the first sentence (placed at a position p) whose normalized score is lower than the ns. Our intuition is that, if a sentence does not exceed the novelty threshold, it is redundant enough and, therefore, a novelty detection mechanism should be applied from this sentence. Conversely, sentences above p have little redundancy and, therefore, they should not be re-ordered using their novelty scores (on the contrary, they should be frozen). Formally, we compute novelty scores for a sentence s_i given its previous ones as:

$$N_{meth_{qdns}}(s_i|s_1, ..., s_{i-1}) = \begin{cases} max_{val} - i & \text{, if } N_{meth_{ns}}(s_i|s_1, ..., s_{i-1}) < ns \\ N_{norm}(s_i|s_1, ..., s_{i-1}) & \text{, otherwise} \end{cases} \tag{2.31}$$

where max_{val} is a large value used to preserve the order of the top $p - 1$ sentences and $N_{meth_{ns}}(s_i|s_1, ..., s_{i-1})$ is[11]:

$$N_{meth_{ns}}(s_i|s_1, ..., s_{i-1}) = \frac{N_{meth}(s_i|s_1, ..., s_{i-1})}{max_q} \tag{2.32}$$

The results obtained are reported in Table 2.14. First, note that the value of the threshold ns tends to be low. This indicates that we start processing novelty only when we find sentences that are much less novel than the most novel sentence in the rank (high redundancy). On the other hand, the overall performance of the variant proposed here outperforms the original novelty detection methods and performs similarly (even slightly higher with NewWords or SetDif in terms of MAP) to the cluster-based variant.

In Figure 2.22 we compare the performance of this variant against the original novelty detection methods, the cluster-based approach and the perfect relevance baseline. The variant proposed here outperforms the baseline, except for SetDif in terms of P@10. In fact, NewWords$_{ns}$ tends to be the approach that supplies the highest performance.

Note that results for the normalized score approach are similar to the results obtained with the cluster-based approach. Since the new score-based variant does not require clustering, it seems that it is an appropriate choice.

2.5 Conclusions

In this chapter we studied different novelty detection approaches and considered two different initial situations: a) a ranking of estimated relevant

[11]Note that, when we consider CosDist, scores have values in [-1 0] and, therefore, to adapt it to the variant proposed here we must sum 1 to CosDist scores.

	NW	**NW**$_{qd_c}$	**NW**$_{qd_{ns}}$	**SD**	**SD**$_{qd_c}$	**SD**$_{qd_{ns}}$	**CD**	**CD**$_{qd_c}$	**CD**$_{qd_{ns}}$
	test: TREC 2003 (train: TREC 2004)								
P@10	.8800	.8960	**.9060**	.8460	.8980*	.8760	.8060	.8920*†	.8780*†
			(ns=0.2)			(ns=0.0)			(ns=0.2)
Δ%		(+1.82)	(+2.95)		(+6.15)	(+3.55)		(+10.67)	(+8.93)
MAP	.8188	.8237	**.8291***†	.7902	.7985	.8113*†	.8046	.8078	.8065
			(ns=0.1)			(ns=0.1)			(ns=0.1)
Δ%		(+0.60)	(+1.26)		(+1.05)	(+2.67)		(+0.40)	(+0.24)
	test: TREC 2004 (train: TREC 2003)								
P@10	.6760	**.7800***†	**.7800***†	.5900	.7700*†	.7540*†	.6460	.7800*†	.7700*†
			(ns=0.2)			(ns=0.1)			(ns=0.6)
Δ%		(+15.38)	(+15.38)		(+30.51)	(+27.80)		(+20.74)	(+19.20)
MAP	.6086	.6389*†	**.6446***†	.5574	.6169*†	.6221*†	.5865	.6340*†	.6303*†
			(ns=0.1)			(ns=0.1)			(ns=0.5)
Δ%		(+4.98)	(+5.92)		(+10.67)	(+11.61)		(+8.10)	(+7.47)

Table 2.14: Comparing query-dependent thresholding (normalized-score and cluster-based) and no-thresholding (original method).

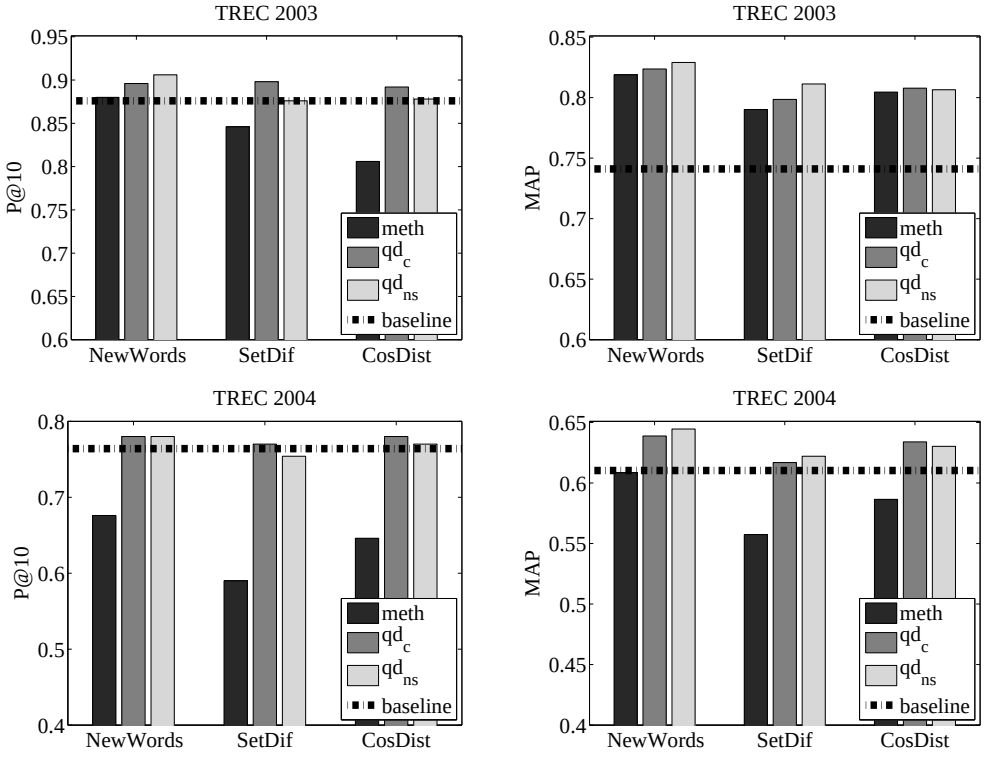


Figure 2.22: Comparison among the best threshold-based approaches proposed here, the corresponding original methods and the perfect baseline.

sentences, i.e. we apply first sentence retrieval in order to generate a ranking

of estimated relevant sentences; and b) a perfect relevance ranking of relevant sentences. The former is a more realistic situation because we do not usually know what sentences are relevant for a query. However, the second permits to study novelty detection without noise coming from non-relevant material. We studied and compared different baselines and considered those that are the strongest.

We evaluated the performance of state of the art novelty detection methods. First, we demonstrated that NewWords performs better than SetDif and CosDist. Next, we designed variants of current novelty methods and proposed new novelty detection mechanisms that perform better than state of the art methods. This was not an easy-to-achieve goal. As a matter of fact, novelty detection at sentence level revealed as an extremely difficult task where naive baselines (e.g. BDOC, which is a straightforward re-ordering of a relevance-oriented ranking) are hardly beaten. Still, we have been able to propose some high performing models.

The variants proposed in order to improve current state of the art novelty detection methods were diverse. On one hand, a simple pruning approach based on considering terms coming from the top 25 sentences for novelty detection purposes helped to improve performance. On the other hand, normalizing by sentence length was also proposed here. However, we concluded that longer sentences tend to be more novel than shorter sentences and, therefore, this mechanism did not help in the novelty estimation.

The use of methods based on smoothed Language Models was also considered here. We studied the impact of smoothing and analyzed, firstly, two different approaches: Aggregate and Non-Aggregate methods (including an efficient version of the later), with two smoothing mechanisms: Dirichlet and Jelinek-Mercer. These are formal methods that outperform the baseline, but the Non-Aggregate versions are the methods that performed the best. The efficient version of the Non-Aggregate Model is not only faster but its performance is similar to the original version. Next, we tested alternative novelty methods based on mixture models that estimate the parameter settings automatically. More specifically, we proposed a two-mixture model and used the Expectation-Maximization algorithm to do automatically the estimation. We found that this approach did not yield to better performance than the baseline.

On the other hand, we demonstrated that the perfect relevance scenario is even more challenging than the non-perfect relevance scenario. In fact, current novelty detection methods are not able to outperform a do-nothing baseline. This happens because, usually, sentences at top positions in the initial ranking tend to be novel and, therefore, they should remain in their original ranking positions. To this aim, we proposed to freeze the top po-

sitions in the initial ranking and followed different approaches in order to determine the position where the novelty detection process should be initiated: a) a query-independent approach, which consists of fixing the same position for all queries; and b) a query-dependent approach, either cluster-based or based on normalizing scores. All these methods are simple and can be computed efficiently. Furthermore, they outperformed significantly the do-nothing baseline. Therefore, this outcome is a novel contribution to the information retrieval community.

Chapter 3

Conclusions

This thesis was centered on sentence retrieval and novelty detection methods. We deeply studied both tasks by analyzing current state of the art methods and proposing efficient and effective alternatives. The conclusions obtained are the following:

- Regarding the performance of current state of the art sentence retrieval models, BM25 is a method that addresses the sentence retrieval problem effectively but it needs parameter tuning. On the other hand, tfisf is a method that performs similarly to tuned BM25 models and it has the advantage of being parameter-free. Therefore, tfisf is a better choice than BM25. However, tfisf is rather ad-hoc (its definition lacks a formal basis).

- We introduced query-independent features into existing retrieval models by following two avenues: a) FLOE, a formal methodology that adjusts directly the sentence weighting by including the values of a given feature; and b) an empirical and simple approximation of the FLOE adjustment. We showed that the adjustment made with empirical methods is usually more consistent than the direct application of FLOE.

- We studied different query-independent features and incorporated them into existing sentence retrieval models. On one hand, we considered opinion-based features: subjectivity, number of positive, negative and opinionated terms. Subjectivity is the feature that, after incorporated into a retrieval model, makes the model to perform the best. The number of negative and opinionated terms also produce benefits. However, the number of positive terms does not provide an added value to these sentence retrieval algorithms.

Named entities are also query-independent features that we have studied. Nevertheless, the incorporation of these named entities does not help to improve the estimation of relevance at sentence level. Usually, named entities are present in queries and, therefore, the explicit incorporation of them as query-independent features is not needed because they are already accounted for when matching sentences and query.

Sentence length is a feature that has been studied thoroughly in document retrieval. We demonstrated that sentence length as a query-independent feature also helps to improve the performance of existing sentence retrieval models. Additionally, in combination with opinion-based features, further gains are obtained.

We believe that opinionated content might be valuable in other classical information retrieval problems and, therefore, we plan to explore this in the near future.

- Sentences are short pieces of information that may result ambiguous or incomplete. In order to avoid this, and given a Language Modeling framework, we incorporated the context into retrieval models by applying localized smoothing. Additionally, we studied the centrality of sentences, i.e. the importance of a sentence in its document. We showed that, after incorporating these two mechanisms, standard sentence retrieval methods perform substantially better.

- We combined sentence retrieval models that use local context with opinion-based features. However, the performance obtained by combining these approaches is not higher than the performance obtained with the local context alone. We analyzed this outcome and found that the local context model is implicitly promoting opinionated information.

- With respect to the local context models, Dirichlet and 2S models with $p(d|s)$ are the methods that perform the best in terms of P@10 and MAP, respectively. With respect to the models based on query-independent features, we considered the subjectivity of a sentence as the best query-independent feature and an empirical method[1] that approximates FLOE as the best approach.

- We considered two possible novelty detection scenarios: perfect relevance and non-perfect relevance. We found that the perfect relevance

[1]Given the subjectivity feature, the behavior of linear, log and step functions is the same.

scenario is harder than the non-perfect relevance scenario (a do-nothing baseline is difficult to beat).

- For the non-perfect relevance scenario we studied two baselines: BNN and BDOC. We showed that BDOC performs better than BNN. In the literature, BNN is often taken as the reference baseline and our comparison between baselines shows that this is misleading.

- NewWords, SetDif and CosDist are current state of the art novelty detection methods. Among them, NewWords is the method that performs the best.

- We proposed variations of standard novelty detection methods (NewWords, SetDif and CosDist). Given a non-perfect relevance scenario, we extracted the terms in the top ranked sentences and used it for vocabulary pruning, i.e. we evaluated novelty by considering only terms in the vocabulary. This variant outperformed the original novelty detection methods. Additionally, we tested the normalization of scores by sentence length. Because long sentences tend to be more relevant than short sentences, this normalization is not appropriate for asymmetric metrics.

- We used Language Models as a tool to estimate novelty. We studied the performance of Non-Aggregate and Aggregate Models and the effect of Dirichlet and Jelinek-Mercer smoothing. In a non-perfect relevance scenario, we demonstrated that Dirichlet smoothing is more robust. Additionally, the Non-Aggregate Model performs better than the Aggregate Model, and outperforms the baseline. Because the Non-Aggregate Model is computationally expensive, we developed a more efficient version that is equally effective than the original model.

  In order to design a novelty detection method that estimates automatically the parameter settings, we adopted a two-mixture model whose parameters are estimated by the Expectation-Maximization algorithm. However, our results indicate that this approach was not as effective as the other Language Models-based methods.

- All the methods proposed for the non-perfect relevance scenario hardly outperform a naive do-nothing baseline in the perfect relevance scenario.

- In a perfect relevance context, we demonstrated that, usually, the top ranked sentences tend to be novel and, therefore, they do not need

to be re-ordered (actually, re-ordering them is harmful). Therefore, we proposed some methods to determine the position where to start applying novelty detection methods. To this aim, we followed query-independent and query-dependent approaches. The former consists of fixing the same position for all queries and starting computing novelty from such position. Regarding query-dependent methods, we followed cluster-based and normalized-score approaches that helped to estimate the position for each query. The query-independent approach outperforms significantly the baseline, and the query-dependent approaches perform slightly higher than the query-independent methods. In fact, among the approaches proposed here, the normalized-score approach is the technique whose performance is the highest.

Appendix A

List of Stopwords

a	a's	able	about	above	according
accordingly	across	actually	after	afterwards	again
against	ain't	all	allow	allows	almost
alone	along	already	also	although	always
am	among	amongst	an	and	another
any	anybody	anyhow	anyone	anything	anyway
anyways	anywhere	apart	appear	appreciate	appropriate
are	aren't	around	as	aside	ask
asking	associated	at	available	away	awfully
b	be	became	because	become	becomes
becoming	been	before	beforehand	behind	being
believe	below	beside	besides	best	better
between	beyond	both	brief	but	by
c	c'mon	c's	came	can	can't
cannot	cant	cause	causes	certain	certainly
changes	clearly	co	com	come	comes
concerning	consequently	consider	considering	contain	containing
contains	corresponding	could	couldn't	course	currently
d	definitely	described	despite	did	didn't
different	do	does	doesn't	doing	don't
done	down	downwards	during	e	each
edu	eg	eight	either	else	elsewhere
enough	entirely	especially	et	etc	even
ever	every	everybody	everyone	everything	everywhere
ex	exactly	example	except	f	far
few	fifth	first	five	followed	following
follows	for	former	formerly	forth	four
from	further	furthermore	g	get	gets
getting	given	gives	go	goes	going
gone	got	gotten	greetings	h	had
hadn't	happens	hardly	has	hasn't	have
haven't	having	he	he's	hello	help
hence	her	here	here's	hereafter	hereby

herein hereupon hers herself hi him
himself his hither hopefully how howbeit
however i i'd i'll i'm i've
ie if ignored immediate in inasmuch
inc indeed indicate indicated indicates inner
insofar instead into inward is isn't
it it'd it'll it's its itself
j just k keep keeps kept
know knows known l last lately
later latter latterly least less lest
let let's like liked likely little
look looking looks ltd m mainly
many may maybe me mean meanwhile
merely might more moreover most mostly
much must my myself n name
namely nd near nearly necessary need
needs neither never nevertheless new next
nine no nobody non none noone
nor normally not nothing novel now
nowhere o obviously of off often
oh ok okay old on once
one ones only onto or other
others otherwise ought our ours ourselves
out outside over overall own p
particular particularly per perhaps placed please
plus possible presumably probably provides q
que quite qv r rather rd
re really reasonably regarding regardless regards
relatively respectively right s said same
saw say saying says second secondly
see seeing seem seemed seeming seems
seen self selves sensible sent serious
seriously seven several shall she should
shouldn't since six so some somebody
somehow someone something sometime sometimes somewhat
somewhere soon sorry specified specify specifying
still sub such sup sure t
t's take taken tell tends th
than thank thanks thanx that that's
thats the their theirs them themselves
then thence there there's thereafter thereby
therefore therein theres thereupon these they
they'd they'll they're they've think third
this thorough thoroughly those though three
through throughout thru thus to together
too took toward towards tried tries
truly try trying twice two u
un under unfortunately unless unlikely until
unto up upon us use used

useful	uses	using	usually	uucp	v
value	various	very	via	viz	vs
w	want	wants	was	wasn't	way
we	we'd	we'll	we're	we've	welcome
well	went	were	weren't	what	what's
whatever	when	whence	whenever	where	where's
whereafter	whereas	whereby	wherein	whereupon	wherever
whether	which	while	whither	who	who's
whoever	whole	whom	whose	why	will
willing	wish	with	within	without	won't
wonder	would	would	wouldn't	x	y
yes	yet	you	you'd	you'll	you're
you've	your	yours	yourself	yourselves	z
zero					

Appendix B

Sentence Retrieval with Localized Smoothing and Sentence Importance

B.1 Localized Smoothing

B.1.1 Training with TREC 2003

	P@10		**MAP**	
BM25	k_1=1.1, b=0, k_3=0		k_1=1.4, b=0, k_3=0	
	$p(q\|s,d)$	$p(q\|s,c_s)$	$p(q\|s,d)$	$p(q\|s,c_s)$
3MM	λ=0.9, γ=0.1	λ=0.9, γ=0.1	λ=0.9, γ=0.1	λ=0.9, γ=0.1
2S	λ=0.4, μ=50	λ=0.2, μ=1	λ=0.6, μ=100	λ=0.1, μ=1
2S-I	λ=0.3, μ=250	λ=0.3, μ=500	λ=0.8, μ=500	λ=0.9, μ=1000
DIR	μ=2500		μ=500	
JM	λ=0.1		λ=0.1	

Table B.1: Optimal parameter settings in the training collection (TREC 2003) for BM25 and LMs without $p(d|s)$.

Context					Document			Surrounding Sents.		
			$p(q\|s)$		$p(q\|s,d)$			$p(q\|s,c_s)$		
	tfisf	*BM25*	*DIR (LMB)*	*JM*	*3MM*	*2S*	*2S-I*	*3MM*	*2S*	*2S-I*
TREC 2002										
P@10	**.2041**	**.2041**†	.1612*	.1163*†	.1122*†	.1265*†	.1918†	.1245*	.1265*†	.1755
Δ% *(tfisf)*		*(+0.00)*	*(-21.02)*	*(-43.02)*	*(-45.03)*	*(-38.02)*	*(-6.03)*	*(-39.00)*	*(-38.02)*	*(-14.01)*
Δ% *(LMB)*	*(+26.61)*	*(+26.61)*		*(-27.85)*	*(-30.40)*	*(-21.53)*	*(+18.98)*	*(-22.77)*	*(-21.53)*	*(+8.87)*
MAP	.1094†	.1102†	.0937*	.0861*†	.0849*	.0938*	**.1218***†	.0837*	.0916*	.1095†
Δ% *(tfisf)*		*(+0.73)*	*(-14.35)*	*(.21.30)*	*(-22.39)*	*(-14.26)*	*(+11.33)*	*(-23.49)*	*(-16.27)*	*(+0.09)*
Δ% *(LMB)*	*(+16.76)*	*(+17.61)*		*(-8.11)*	*(-9.39)*	*(+0.11)*	*(+29.99)*	*(-10.67)*	*(-2.24)*	*(+16.86)*
TREC 2004										
P@10	.4300	.4380	.4020	.3580*†	.3560*†	.3220*†	.4660*†	.3260*†	.3420*†	**.4760***†
Δ% *(tfisf)*		*(+1.86)*	*(-6.51)*	*(-16.74)*	*(-17.21)*	*(-25.12)*	*(+8.37)*	*(-24.19)*	*(-20.47)*	*(+10.70)*
Δ% *(LMB)*	*(+6.97)*	*(+8.96)*		*(-10.95)*	*(-11.44)*	*(-19.90)*	*(+15.92)*	*(-18.91)*	*(-14.93)*	*(+18.41)*
MAP	.2358†	.2368*†	.2240*	.2131*†	.2199*	.2204*	**.2607***†	.2124*†	.2204*	.2496*†
Δ% *(tfisf)*		*(+0.42)*	*(-5.00)*	*(-9.63)*	*(-6.74)*	*(-6.53)*	*(+10.56)*	*(-9.92)*	*(-6.53)*	*(+5.85)*
Δ% *(LMB)*	*(+5.27)*	*(+5.71)*		*(-4.87)*	*(-1.83)*	*(-1.61)*	*(+16.38)*	*(-5.18)*	*(-1.61)*	*(+11.43)*

Table B.2: P@10 and MAP in the test collections (TREC 2002 & TREC 2004). Statistically significant differences with respect to tfisf are marked with * and with respect to LMB are marked with †.

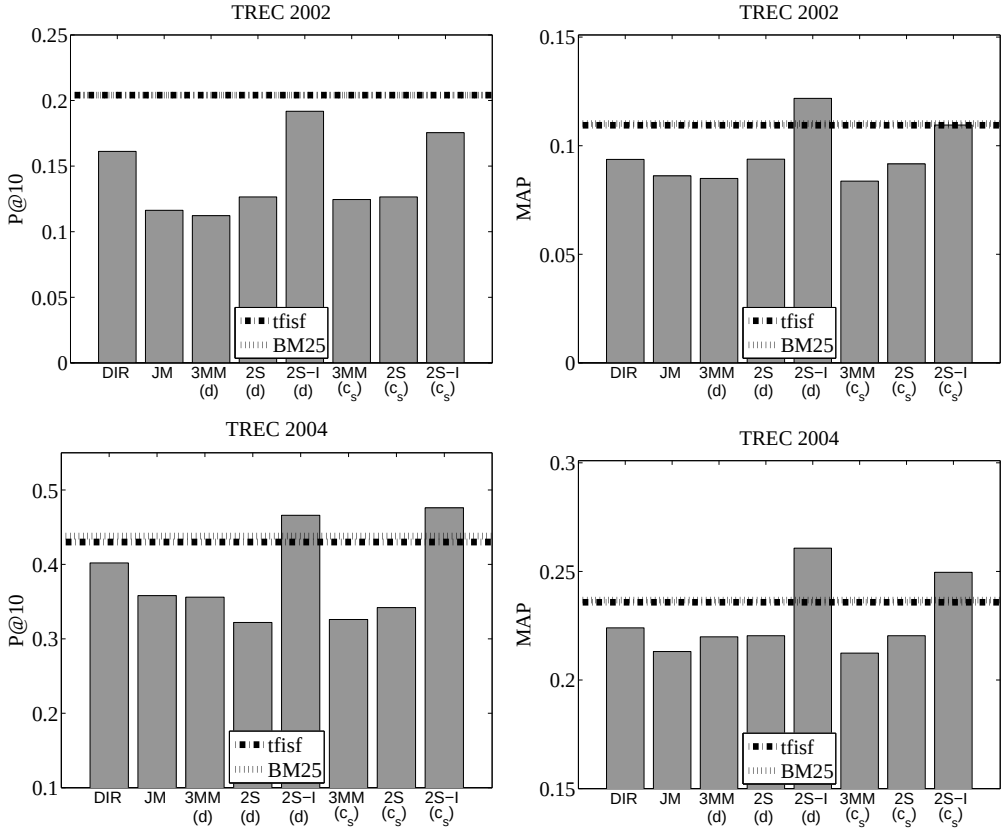


Figure B.1: P@10 and MAP in the test collections (TREC 2002 & TREC 2004) without sentence importance.

B.1.2 Training with TREC 2004

	P@10		MAP	
BM25	k_1=1.0, b=0, k_3=0		k_1=1.0, b=0, k_3=0	
	$p(q\|s,d)$	$p(q\|s,c_s)$	$p(q\|s,d)$	$p(q\|s,c_s)$
3MM	λ=0.8, γ=0.1	λ=0.8, γ=0.1	λ=0.9, γ=0.1	λ=0.8, γ=0.1
2S	λ=0.8, μ=10000	λ=0.2, μ=1	λ=0.1, μ=250	λ=0.1, μ=1
2S-I	λ=0.6, μ=250	λ=0.4, μ=500	λ=0.8, μ=100	λ=0.7, μ=500
DIR	μ=250		μ=500	
JM	λ=0.1		λ=0.1	

Table B.3: Optimal parameter settings in the training collection (TREC 2004) for BM25 and LMs without $p(d|s)$.

Context					**Document**			**Surrounding Sents.**		
			$p(q\|s)$		$p(q\|s,d)$			$p(q\|s,c_s)$		
	tfisf	*BM25*	*DIR (LMB)*	*JM*	*3MM*	*2S*	*2S-I*	*3MM*	*2S*	*2S-I*
TREC 2002										
P@10	.2041	.2041†	.1633*	.1163*†	.1061*†	.1531*	**.2245**†	.1286*†	.1265*†	.1837
Δ% (tfisf)		*(+0.00)*	*(-19.99)*	*(-43.02)*	*(-48.02)*	*(-24.99)*	*(+10.00)*	*(-36.99)*	*(-38.02)*	*(-10.00)*
Δ% (LMB)	*(+24.98)*	*(+24.98)*		*(-28.78)*	*(-35.03)*	*(-6.25)*	*(+37.48)*	*(-21.25)*	*(-22.54)*	*(+12.49)*
MAP	.1094†	.1102†	.0937*	.0861*†	.0849*	.0917*	**.1200**†	.0919*	.0916*	.1096†
Δ% (tfisf)		*(+0.73)*	*(-14.35)*	*(-21.30)*	*(-22.39)*	*(-16.18)*	*(+9.69)*	*(-16.00)*	*(-16.27)*	*(+0.18)*
Δ% (LMB)	*(+16.76)*	*(+17.61)*		*(-8.11)*	*(-9.39)*	*(-2.13)*	*(+28.07)*	*(-1.92)*	*(-2.24)*	*(+16.97)*
TREC 2003										
P@10	.7480	.7520†	.7140*	.5600*†	.5480*†	.5800*†	.7400	.5400*†	.5320*†	**.7540**†
Δ% (tfisf)		*(+0.53)*	*(-4.55)*	*(-25.13)*	*(-26.74)*	*(-22.46)*	*(-1.07)*	*(-27.81)*	*(-28.88)*	*(+0.80)*
Δ% (LMB)	*(+4.76)*	*(+5.32)*		*(-21.56)*	*(-23.25)*	*(-18.77)*	*(+3.64)*	*(-24.37)*	*(-25.49)*	*(+5.60)*
MAP	.3851†	.3846†	.3638*	.3474*†	.3555*	.3503*	**.4098***†	.3532*†	.3494*†	.3900†
Δ% (tfisf)		*(-0.13)*	*(-5.53)*	*(-9.79)*	*(-7.69)*	*(-9.03)*	*(+6.41)*	*(-8.28)*	*(-9.27)*	*(+1.27)*
Δ% (LMB)	*(+5.85)*	*(+5.72)*		*(-4.51)*	*(-2.28)*	*(-3.71)*	*(+12.64)*	*(-2.91)*	*(-3.96)*	*(+7.20)*

Table B.4: P@10 and MAP in the test collections (TREC 2002 & TREC 2003). Statistically significant differences with respect to tfisf are marked with * and with respect to LMB are marked with †.

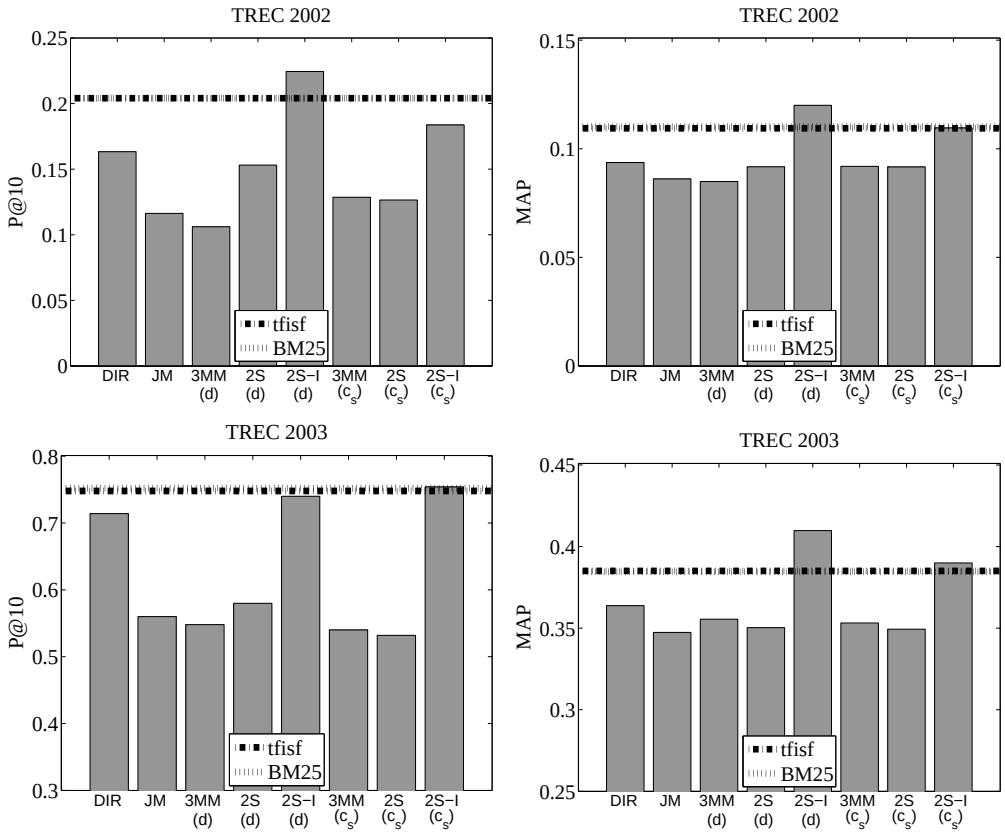


Figure B.2: P@10 and MAP in the test collections (TREC 2002 & TREC 2003) without sentence importance.

B.2 Sentence Importance

B.2.1 Training with TREC 2003

	P@10		MAP	
BM25	k_1=1.1, b=0, k_3=0		k_1=1.4, b=0, k_3=0	
	$p(q\|s,d)p(d\|s)$	$p(q\|s,c_s)p(d\|s)$	$p(q\|s,d)p(d\|s)$	$p(q\|s,c_s)p(d\|s)$
3MM	λ=0.6, γ=0.1	λ=0.7, γ=0.1	λ=0.8, γ=0.1	λ=0.8, γ=0.1
2S	λ=0.1, μ=1	λ=0.1, μ=1	λ=0.1, μ=1	λ=0.1, μ=1
2S-I	λ=0.1, μ=10	λ=0.1, μ=5	λ=0.1, μ=1	λ=0.1, μ=1
DIR	μ=1		μ=1	
JM	λ=0.1		λ=0.1	

Table B.5: Optimal parameter settings in the training collection (TREC 2003) for LMs with $p(d|s)$.

Context	**Sentence Only**				**Document**			**Surrounding Sents.**		
			$p(q\|s)p(d\|s)$		$p(q\|s,d)p(d\|s)$			$p(q\|s,c_s)p(d\|s)$		
	tfisf	*BM25*	*DIR*	*JM*	*3MM*	*2S*	*2S-I*	*3MM*	*2S*	*2S-I*
TREC 2002										
P@10	.2041†	.2041†	2429†	.2449†	2429†	**.2469**†	.2429†	2449†	.2449†	.2449†
Δ% (tfisf)		*(+0.00)*	*(+19.01)*	*(+19.99)*	*(+19.01)*	*(+20.97)*	*(+19.01)*	*(+19.99)*	*(+19.99)*	*(+19.99)*
Δ% (LMB)	*(+26.61)*	*(+26.61)*	*(+50.68)*	*(+51.92)*	*(+50.68)*	*(+53.16)*	*(+50.68)*	*(+51.92)*	*(+51.92)*	*(+51.92)*
MAP	.1094†	.1102†	**.1349***†	.1347*†	.1333*†	.1344*†	.1329*†	.1342*†	.1343*†	.1347*†
Δ% (tfisf)		*(+0.73)*	*(+23.31)*	*(+23.13)*	*(+21.85)*	*(+22.85)*	*(+21.48)*	*(+22.67)*	*(+22.76)*	*(+23.13)*
Δ% (LMB)	*(+16.76)*	*(+17.61)*	*(+43.97)*	*(+43.76)*	*(+42.26)*	*(+43.44)*	*(+41.84)*	*(+43.22)*	*(+43.33)*	*(+43.76)*
TREC 2004										
P@10	.4300	.4380	.4420	**.4480**	.4400	.4420	.4360	.4460	.4400	.4440
Δ% (tfisf)		*(+1.86)*	*(+2.79)*	*(+4.19)*	*(+2.33)*	*(+2.79)*	*(+1.40)*	*(+3.72)*	*(+2.33)*	*(+3.26)*
Δ% (LMB)	*(+6.97)*	*(+8.96)*	*(+9.95)*	*(+11.44)*	*(+9.45)*	*(+9.95)*	*(+8.46)*	*(+10.95)*	*(+9.45)*	*(+10.45)*
MAP	.2358†	.2368*†	2549*†	.2548*†	2531*†	.2538*†	.2532*†	2550*†	.2551*†	**.2553***†
Δ% (tfisf)		*(+0.42)*	*(+8.10)*	*(+8.06)*	*(+7.34)*	*(+7.63)*	*(+7.38)*	*(+8.14)*	*(+8.18)*	*(+8.27)*
Δ% (LMB)	*(+5.27)*	*(+5.71)*	*(+13.79)*	*(+13.75)*	*(+12.99)*	*(+13.30)*	*(+13.04)*	*(+13.84)*	*(+13.88)*	*(+13.97)*

Table B.6: P@10 and MAP in the test collections (TREC 2002 & TREC 2004). Statistically significant differences with respect to tfisf are marked with * and with respect to standard DIR (LMB) are marked with †.

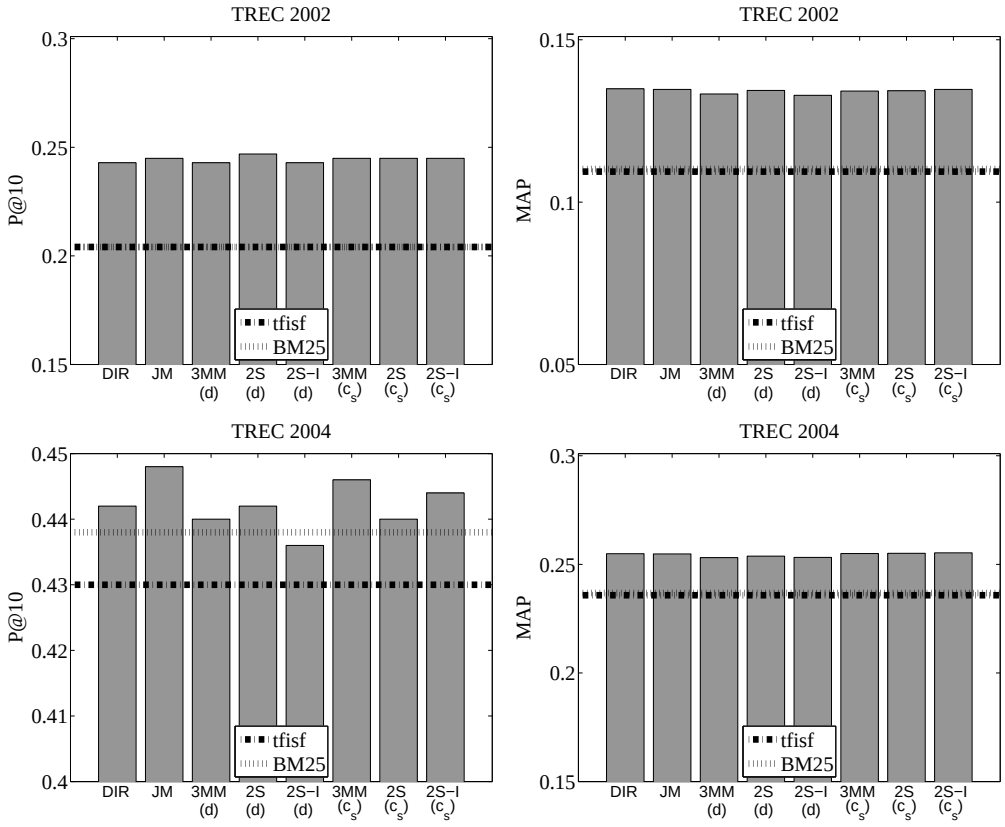


Figure B.3: P@10 and MAP in the test collections (TREC 2002 & TREC 2004) with *p(d|s)*.

B.2.2 Training with TREC 2004

	P@10		MAP	
BM25	k_1=1.0, b=0, k_3=0		k_1=1.0, b=0, k_3=0	
	$p(q\|s,d)p(d\|s)$	$p(q\|s,c_s)p(d\|s)$	$p(q\|s,d)p(d\|s)$	$p(q\|s,c_s)p(d\|s)$
3MM	λ=0.9, γ=0.1	λ=0.4, γ=0.4	λ=0.8, γ=0.1	λ=0.4, γ=0.5
2S	λ=0.1, μ=1	λ=0.2, μ=25	λ=0.1, μ=1	λ=0.1, μ=1
2S-I	λ=0.2, μ=5	λ=0.3, μ=5	λ=0.1, μ=1	λ=0.1, μ=1
DIR	μ=5		μ=1	
JM	λ=0.1		λ=0.1	

Table B.7: Optimal parameter settings in the training collection (TREC 2004) for LMs with $p(d|s)$.

Context	**Sentence Only**				**Document**			**Surrounding Sents.**		
			$p(q\|s)p(d\|s)$		$p(q\|s,d)p(d\|s)$			$p(q\|s,c_s)p(d\|s)$		
	tfisf	*BM25*	*DIR*	*JM*	*3MM*	*2S*	*2S-I*	*3MM*	*2S*	*2S-I*
					TREC 2002					
P@10	.2041†	.2041†	.2449†	.2449†	.1796	**.2469**†	**.2469**†	.2449†	.2449†	.2449†
Δ% *(tfisf)*		*(+0.00)*	*(+19.99)*	*(+19.99)*	*(-12.00)*	*(+20.97)*	*(+20.97)*	*(+19.99)*	*(+19.99)*	*(+19.99)*
Δ% *(LMB)*	*(+24.98)*	*(+24.98)*	*(+49.97)*	*(+49.97)*	*(+9.98)*	*(+51.19)*	*(+51.19)*	*(+49.97)*	*(+49.97)*	*(+49.97)*
MAP	.1094†	.1102†	**.1349***†	.1347*†	.1333*†	.1344*†	.1329*†	.1344*†	.1343*†	.1347*†
Δ% *(tfisf)*		*(+0.73)*	*(+23.31)*	*(+23.13)*	*(+21.85)*	*(+22.85)*	*(+21.48)*	*(+22.85)*	*(+22.76)*	*(+23.13)*
Δ% *(LMB)*	*(+16.76)*	*(+17.61)*	*(+43.97)*	*(+43.76)*	*(+42.26)*	*(+43.44)*	*(+41.84)*	*(+43.44)*	*(+43.33)*	*(+43.76)*
					TREC 2003					
P@10	.7480†	**.7520**†	.7500	.7480	.6960	.7440	.7360	.7360	.7360	.7420
Δ% *(tfisf)*		*(+0.53)*	*(+0.27)*	*(+0.00)*	*(-6.95)*	*(-0.53)*	*(-1.60)*	*(-1.60)*	*(-1.60)*	*(-0.80)*
Δ% *(LMB)*	*(+4.76)*	*(+5.32)*	*(+5.04)*	*(+4.76)*	*(-2.52)*	*(+4.20)*	*(+3.08)*	*(+3.08)*	*(+3.08)*	*(+3.92)*
MAP	.3851†	.3846†	**.4144***†	.4137*†	.4111*†	.4117*†	.4113*†	.4126*†	.4135*†	.4139*†
Δ% *(tfisf)*		*(-0.13)*	*(+7.61)*	*(+7.43)*	*(+6.75)*	*(+6.91)*	*(+6.80)*	*(+7.14)*	*(+7.37)*	*(+7.48)*
Δ% *(LMB)*	*(+5.85)*	*(+5.72)*	*(+13.91)*	*(+13.72)*	*(+13.00)*	*(+13.17)*	*(+13.06)*	*(+13.41)*	*(+13.66)*	*(+13.77)*

Table B.8: P@10 and MAP in the test collections (TREC 2003 & TREC 2004) after incorporating sentence importance ($p(d|s)$). Statistically significant differences with respect to tfisf are marked with * and with respect to standard DIR (LMB) are marked with †.

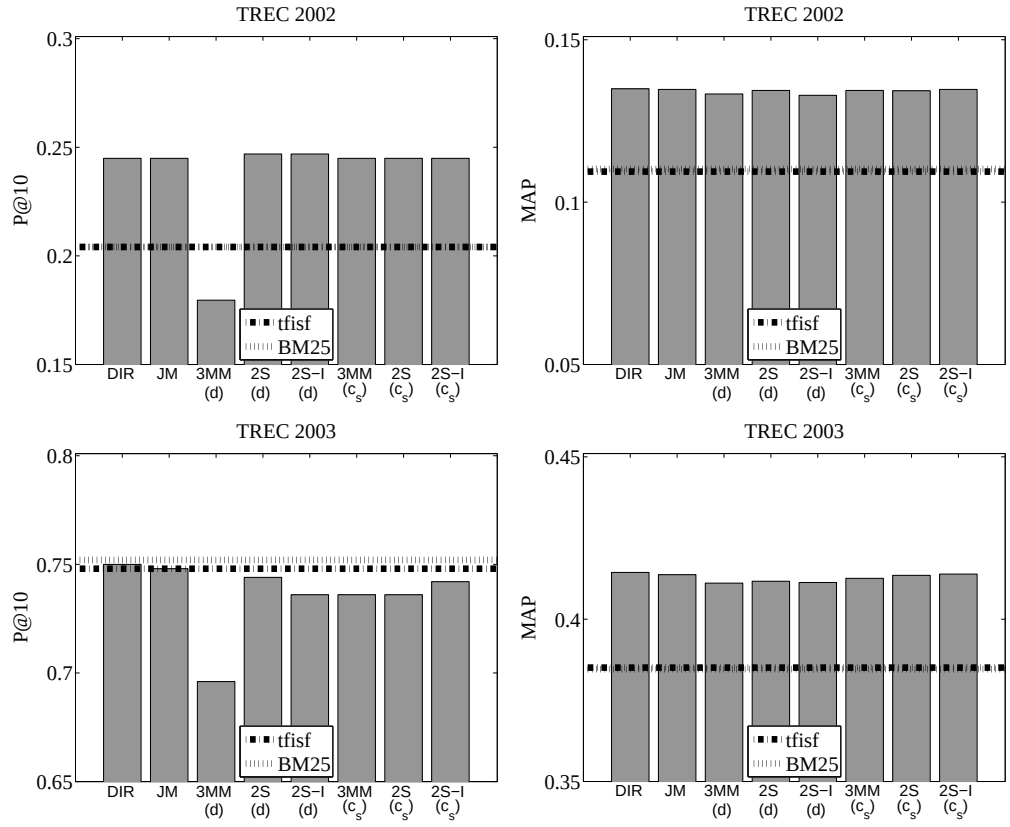


Figure B.4: P@10 and MAP in the test collections (TREC 2002 & TREC 2003) with *p(d|s)*.

Appendix C

Expectation-Maximization Algorithm

The Expectation-Maximization (EM) algorithm [DLR77, MK08] is a general method to find the maximum-likelihood estimate of the parameters of an underlying distribution from a given data set when the data is incomplete or missing values.

Let $\mathcal{X}$ be the set of observed data, that is, the set of known elements or samples. This set is generated following some kind of probabilistic distribution.

We define $\mathcal{Y}$ as the set of unobserved data, that is, the set of unknown or hidden variables which make influence over the generation of $\mathcal{X}$.

We also define $\mathcal{Z}$ as a complete data set composed by $\mathcal{X}$ and $\mathcal{Y}$. That is, $\mathcal{Z} = (\mathcal{X}, \mathcal{Y})$.

We assume the following density function for any data in $\mathcal{Z}$:

$$p(\mathbf{z}|\Theta) = p(\mathbf{x}, \mathbf{y}|\Theta) = p(\mathbf{y}|\mathbf{x}, \Theta) \cdot p(\mathbf{x}|\Theta) \tag{C.1}$$

That means that if we want to calculate the probability of a data pair (x, y) given a parameter configuration (Θ), we can compute it as the probability of any of the data (y) conditioned to the other data (x) and the parameters (Θ) [1].
Assuming these definitions, we can divide the EM algorithm in two steps:

(i) **E-step**: In this step it finds the expected value for the complete data

[1] In the same way, we could have written the expression above as:

$$p(\mathbf{z}|\Theta) = p(\mathbf{x}, \mathbf{y}|\Theta) = p(\mathbf{x}|\mathbf{y}, \Theta) \cdot p(\mathbf{y}|\Theta) \tag{C.2}$$

log-likelihood with respect to the unknown data $\mathcal{Y}$ given the observed data $\mathcal{X}$ and the current parameter estimates.

In this step we define Q as:

$$Q(\Theta, \Theta^{(i-1)}) = E\Big[\log p\big(\mathcal{X}, \mathcal{Y}|\Theta\big)|\mathcal{X}, \Theta^{(i-1)}\Big] \tag{C.3}$$

That is, given the observed data ($\mathcal{X}$) and the previous parameter setting ($\Theta^{(i-1)}$), we compute the complete data log-likelihood (starting the process with an initial and random parameter setting, Θ^0). The idea is that, since we do not know the values of $\mathcal{Y}$, we assume that $\mathcal{Y}$ is a random variable governed by a probability distribution and we compute the expectation over each possible value $y_i \in \Upsilon$.

In the continuous case, the expression is:

$$E\Big[\log p\big(\mathcal{X}, \mathcal{Y}|\Theta\big)|\mathcal{X}, \Theta^{(i-1)}\Big] = \int_{\mathbf{y}\in\Upsilon} \log p(\mathcal{X}, \mathbf{y}|\Theta) \cdot f(\mathbf{y}|\mathcal{X}, \Theta^{(i-1)}) d\mathbf{y} \tag{C.4}$$

and, in the discrete case:

$$E\Big[\log p\big(\mathcal{X}, \mathcal{Y}|\Theta\big)|\mathcal{X}, \Theta^{(i-1)}\Big] = \sum_{\mathbf{y}\in\Upsilon} \log p(\mathcal{X}, \mathbf{y}|\Theta) \cdot f(\mathbf{y}|\mathcal{X}, \Theta^{(i-1)}) \tag{C.5}$$

In this case, Υ indicates the range of possible values for $y \in \mathcal{Y}$. In any case, $f(\mathbf{y}|\mathcal{X}, \Theta^{(i-1)})$ is the marginal distribution of the unobserved data and it is dependent on both the observed data and the parameters.

(ii) **M-step**: This step maximizes the expectation we computed in the *E-step*, that is, we estimate the new parameters in the iteration i as:

$$\Theta^{(i)} = arg\ max_{\Theta} Q(\Theta, \Theta^{(i-1)}) \tag{C.6}$$

The EM algorithm guarantees that it always converges to a local maximum of the likelihood function.

C.1 Expectation-Maximization Algorithm for Estimating Mixture Model's Parameters

The mixture model computes the probability of an element $\mathbf{x}$ as:

$$p(\mathbf{x}|\Theta) = \sum_{i=1}^{M} \alpha_i \cdot p_i(\mathbf{x}|\theta_i) \tag{C.7}$$

where M is the number of probabilistic models, $\{\alpha_1, ..., \alpha_M\}$ are the weighting parameter values associated with each probability distribution and $\{\theta_1, ..., \theta_M\}$ are the parameters related to each probabilistic model. It is always verified that $\sum_{i=1}^{M} \alpha_i = 1$.

The expression above means that samples are generated following M distinct density functions (that is, each sample x_i belongs to one of these M density functions). Within this generation process, each density function has a weight which is defined by α_i.

Given a data set $\mathcal{X}$, generated from this mixture model, the density function of the data set $\mathcal{X}$ (of size N) and the likelihood expression for this set is:

$$\mathcal{L}(\Theta|\mathcal{X}) = p(\mathcal{X}|\Theta) = \prod_{i=1}^{N} p(x_i|\Theta) \tag{C.8}$$

which, applying logarithms and using Equation C.7, we have that:

$$\log \prod_{i=1}^{N} p(x_i|\Theta) = \sum_{i=1}^{N} \log p(x_i|\Theta) = \sum_{i=1}^{N} \log \Big(\sum_{j=1}^{M} \alpha_j \cdot p_j(x_i|\theta_j) \Big) \tag{C.9}$$

Now, let us introduce the $\mathcal{Y}$ variables. This set of variables will represent the hidden variables. For each $x_i \in \mathcal{X}$ ($\{x_1, ..., x_n\}$) there is a hidden variable y_i which indicates the distribution that generated the sample x_i[2]. The unobserved data elements are therefore $\mathcal{Y} = \{y_i\}_{i=1}^{N}$, where the element y_i has the value k if the element x_i was generated by the k^{th} mixture component. The range of possible values for the elements in $\mathcal{Y}$ is $\{1, ..., M\}$.

Given these hidden variables, the second addition in the expression in the Equation C.9 is converted into single addend, which takes the value $y_i = k$ when k is the mixture component which generated the element x_i:

$$\log(\mathcal{L}(\Theta|\mathcal{X}, \mathcal{Y})) = \sum_{i=1}^{N} \log \Big(\sum_{j=1}^{M} \alpha_j \cdot p_j(x_i|\theta_j) \Big) = \sum_{i=1}^{N} \log(\alpha_{y_i} \cdot p_{y_i}(x_i|\theta_{y_i})) \tag{C.10}$$

In practice, we do not know the values of $\mathcal{Y}$ but we can assume that the elements in $\mathcal{Y}$ have random values and follow a given probability distribution.

Let us suppose that we have some initial values for the parameters α_j and θ_j. We will call those parameter values as $\Theta^g = (\alpha_1^g, ..., \alpha_M^g, \theta_1^g, ..., \theta_M^g)$.

[2]The complete data set $\mathcal{Z}$ will be composed by N samples (since $|\mathcal{X}| = N$) and N variables (since $|\mathcal{Y}| = N$) and, so, the cardinality of $\mathcal{Z}$ is $2N$.

These parameters can be initialized randomly. So, given Θ^g, we can compute $p(x_i|\Theta^g)$:

$$p(x_i|\Theta^g) = \sum_{j=1}^{M} \alpha_j^g \cdot p_j(x_i|\theta_j^g) \tag{C.11}$$

The mixing parameters α_j can be thought as prior probabilities of each mixture component, that is, $\alpha_j = p(component_j)$

We can extract from the Equation C.10 the expressions:

$$p(x_i|y_i) = p_{y_i}(x_i|\theta_{y_i}) \tag{C.12}$$

$$p(y_i) = \alpha_{y_i} \tag{C.13}$$

When the variable y_i is known, we also know the distribution p_{y_i} which generated the sample x_i.

To compute $p(y_i|x_i, \Theta^g)$ we apply the Bayes' rule:

$$p(y_i|x_i, \Theta^g) = \frac{p(x_i|y_i, \Theta^g) \cdot p(y_i)}{p(x_i|\Theta^g)} \tag{C.14}$$

From Equations C.11, C.12 and C.13 we have that:

$$p(y_i|x_i, \Theta^g) = \frac{p(x_i|y_i, \Theta^g) \cdot p(y_i)}{p(x_i|\Theta^g)} = \frac{\alpha_{y_i}^g \cdot p_{y_i}(x_i|\theta_{y_i}^g)}{\sum_{j=1}^{M} \alpha_j^g \cdot p_j(x_i|\theta_k^g)} \tag{C.15}$$

Moreover, we know that:

$$p(\mathbf{y}|\mathcal{X}, \Theta^g) = \prod_{i=1}^{N} p(y_i|x_i, \Theta^g) \tag{C.16}$$

where $\mathbf{y} = \{y_1, ..., y_N\}$

Let us take again the Equation C.5. In our case, f is a probability function, $\Theta^{i-1} = \Theta^g$:

$$Q(\Theta, \Theta^g) = \sum_{\mathbf{y} \in \Upsilon} \log(\mathcal{L}(\Theta|\mathcal{X}, \mathbf{y})) \cdot p(\mathbf{y}|\mathcal{X}, \Theta^g) \tag{C.17}$$

If we substitute each expression for the equivalences given in the Equations C.10 and C.16, we have:

$$\sum_{\mathbf{y} \in \Upsilon} \sum_{i=1}^{N} \log(\alpha_{y_i} \cdot p_{y_i}(x_i|\theta_{y_i})) \prod_{j=1}^{N} p(y_j|x_j, \theta^g) \tag{C.18}$$

Since we do not know which distribution generated each sample, we need to try every possible combination. The possible values for every $y_i \in \Upsilon$ are the number of probabilistic models we have, that is, $\{1, ..., M\}$. Moreover, we need to do that for every y_i where $i = \{1, ..., N\}$. So, we can translate the expression above to:

$$\sum_{y_1=1}^{M}\sum_{y_2=1}^{M}...\sum_{y_N=1}^{M}\sum_{j=1}^{N}\log(\alpha_{y_i}\cdot p_{y_i}(x_i|\theta_{y_i}))\prod_{j=1}^{N}p(y_j|x_j,\theta^g) \tag{C.19}$$

The problem now is that we do not know which distribution generated the sample x_i and, so, we have to evaluate the expression above with every possible values of y_i, that is, from 1 to M:

$$\sum_{y_1=1}^{M}\sum_{y_2=1}^{M}...\sum_{y_N=1}^{M}\sum_{i=1}^{N}\sum_{\ell=1}^{M}\delta_{\ell,y_i}\cdot\log(\alpha_\ell\cdot p_\ell(x_i|\theta_\ell))\cdot\prod_{j=1}^{N}p(y_j|x_j,\theta^g) \tag{C.20}$$

which is the same as:

$$Q(\Theta,\Theta^g)=\sum_{\ell=1}^{M}\sum_{i=1}^{N}\log(\alpha_\ell\cdot p_\ell(x_i|\theta_\ell))\cdot\sum_{y_1=1}^{M}\sum_{y_2=1}^{M}...\sum_{y_N=1}^{M}\delta_{\ell,y_i}\cdot\prod_{j=1}^{N}p(y_j|x_j,\Theta^g) \tag{C.21}$$

Because $y_i = \ell$ and $\sum_{y_i=1}^{M} p(y_i|x_i,\Theta^g) = 1$, this expression can be simplified as:

$$\sum_{y_1=1}^{M}\sum_{y_2=1}^{M}...\sum_{y_N=1}^{M}\delta_{\ell,y_i}\cdot\prod_{j=1}^{N}p(y_j|x_j,\Theta^g)=$$

$$\Big(\sum_{y_1=1}^{M}...\sum_{y_{i-1}=1}^{M}\sum_{y_{i+1}=1}^{M}...\sum_{y_N=1}^{M}\prod_{j=1,j\neq i}p(y_j|x_j,\Theta^g)\Big)\cdot p(\ell|x_i,\Theta^g) \tag{C.22}$$

We can rewrite this expression as follows:

$$\prod_{j=1,j\neq i}^{N}\Big(\sum_{y_j=1}^{M}p(y_j|x_j,\Theta^g)\Big)\cdot p(\ell|x_i,\Theta^g) \tag{C.23}$$

Applying the property $\sum_{y_i=1}^{M} p(y_i|x_i,\Theta^g) = 1$ we have that:

$$\prod_{j=1,j\neq i}\Big(\sum_{y_j=1}^{M}p(y_j|x_j,\Theta^g)\Big)=1 \tag{C.24}$$

and, thus, the expression above rests simplified as:

$$\sum_{y_1=1}^{M}\sum_{y_2=1}^{M}...\sum_{y_N=1}^{M}\delta_{\ell,y_i}\cdot\prod_{j=1}^{N}p(y_j|x_j,\Theta^g)=p(\ell|x_i,\Theta^g) \tag{C.25}$$

Applying this equality to Equation C.21 we obtain:

$$Q(\Theta,\Theta^g)=\sum_{\ell=1}^{M}\sum_{j=1}^{N}log(\alpha_\ell\cdot p_\ell(x_i|\theta_\ell))p(\ell|x_i,\Theta^g) \tag{C.26}$$

Applying the product property for logarithms:

$$Q(\Theta,\Theta^g)=\sum_{\ell=1}^{M}\sum_{j=1}^{N}\log(\alpha_\ell)\cdot p(\ell|x_i,\Theta^g)+\sum_{\ell=1}^{M}\sum_{j=1}^{N}\cdot\log(p_\ell(x_i|\theta_\ell))\cdot p(\ell|x_i,\Theta^g) \tag{C.27}$$

We need to find the expression for α_ℓ.

To simplify, we will use Lagrange multiplier λ, considering that $\sum_\ell \alpha_\ell = 1$. We use the partial derivative of α:

$$\frac{\partial}{\partial\alpha_\ell}\Big[\sum_{\ell=1}^{M}\sum_{i=1}^{N}\log(\alpha_\ell)\cdot p(\ell|x_i,\Theta^g)+\lambda\cdot\Big(\sum_{\ell}\cdot\alpha_\ell-1\Big)\Big]=0 \tag{C.28}$$

We obtain the following:

$$\sum_{i=1}^{N}\frac{1}{N}\sum_{i=1}^{N}p(\ell|x_i,\Theta^g) \tag{C.29}$$

If we solve this expression, we get that $\lambda=-N$. So, we get the value for α_ℓ:

$$\alpha_\ell=\frac{1}{N}\sum_{i=1}^{N}p(\ell|x_i,\Theta^g) \tag{C.30}$$

Assuming that ℓ is every possible value for y_i (that is, $\{1,...,M\}$), we can rewrite this expression for each y_i as:

$$\alpha_{y_i}=\frac{1}{N}\sum_{j=1}^{N}p(y_i|x_j,\Theta^g) \tag{C.31}$$

So, given the samples x_j and a parameter setting Θ^g, we compute α_{y_i} as the average between the probabilities of every possible value for the hidden variable y_i given x_i and Θ^g.

The value obtained for this parameter α_{y_i} is the one which satisfies the condition $\Theta^{(i)} = arg\ max_{\Theta} Q(\Theta, \Theta^{(i-1)})$ in the M-step of the EM algorithm. We adjusted the α_{y_i} scores (which reflect the relative importance of the distributions) using Equation C.31.

It would be possible to have parameters associated to the distributions whose value is unknown (e.g. mean and covariance) but this does not happen in our case and, so, we consider unnecessary to introduce them in this work.

To sum up, the key expressions for each step of the EM algorithm in this context have been reduced to:

(1) **E-Step**:

$$p(y_i|x_i, \Theta^g) = \frac{\alpha^g_{y_i} \cdot p_{y_i}(x_i|\theta^g_{y_i})}{\sum_{k=1}^{M} \alpha^g_k \cdot p_k(x_i|\theta^g_k)} \tag{C.32}$$

(2) **M-Step**:

$$\alpha_{y_i} = \frac{1}{N} \sum_{j=1}^{N} p(y_i|x_j, \Theta^g) \tag{C.33}$$

Given these generic expressions, the next stage is adapting them to our particular domain.

Appendix D

Novelty Detection Results with a Perfect Relevance Baseline

In Section 2.2 we used a baseline to estimate the relevance of sentences and, next, re-ordered them following their novelty scores. The final novelty ranking was dependent, therefore, on both the quality of the sentence retrieval and novelty detection methods. Although this approach is realistic, it is not the most appropriate way to study the performance of novelty detection mechanisms because the sentence retrieval step may introduce some noise[1]. Furthermore, the order followed by assessors and the novelty detection system will be different. The order of the relevance ranking directly affects the performance of a novelty detection system because the novelty nature of sentences in the ranking is dependent on the set of previously seen sentences. To address this problem, in this section we consider the relevance judgments taken from the TREC 2003 and 2004 datasets. This ranking corresponds to the ranked list of sentences for each query that assessors judged as relevant to the given query. Considering this ranking as the input to the novelty detection processes, we can study novelty independently of relevance. In next sections we report the effectiveness of the methods presented in Section 2.2 considering the perfect relevance scenario.

[1]The final novelty ranking consists of a set of sentences that are both relevant and novel. If the sentence retrieval method we used does not work properly, the final ranking will not be good enough (regardless of novelty detection's performance).

D.1 Standard Novelty Detection Methods

	baseline	NewWords	SetDif	CosDist
		TREC 2003		
P@10	.8760	**.8800**	.8460	.8060
$\Delta\%$		$(+0.46)$	(-3.42)	(-7.99)
MAP	.7411	**.8188*†**	.7902	.8046*
$\Delta\%$		$(+10.48)$	$(+6.63)$	$(+8.57)$
		TREC 2004		
P@10	**.7640**	.6760*	.5900*†	.6460*†
$\Delta\%$		(-11.52)	(-22.77)	(-15.44)
MAP	**.6103**	.6086	.5574	.5865
$\Delta\%$		(-0.28)	(-8.67)	(-3.90)

Table D.1: NewWords, SetDif and CosDist performance against the perfect relevance baseline.

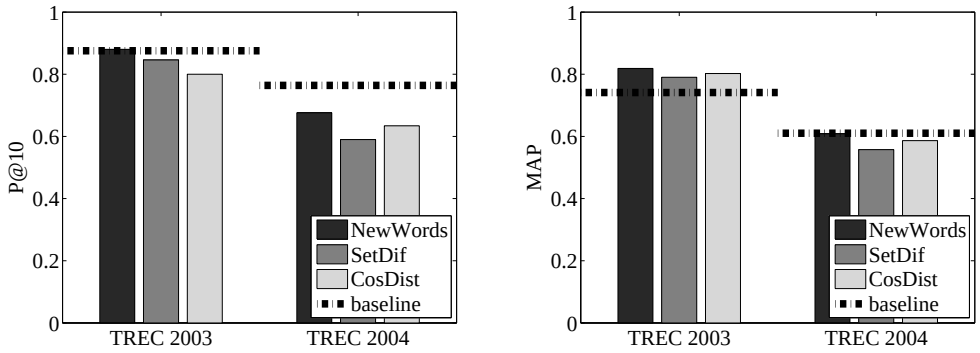


Figure D.1: NewWords, SetDif and CosDist performance against the perfect relevance baseline.

D.2 Normalized Standard Novelty Detection Methods

	NewWords	NewWords$_n$	SetDif	SetDif$_n$	CosDist	CosDist$_n$
			TREC 2003			
P@10	**.8800**	.8360*	**.8460**	.8300	.8060	**.8360***
$\Delta\%$		(-5.00)		(-1.89)		$(+3.72)$
MAP	**.8188**	.8121	.7902	**.8043***	.8046	**.8104**
$\Delta\%$		(-0.82)		$(+1.78)$		$(+0.72)$
			TREC 2004			
P@10	.6760	**.6960**	.5900	**.6600***	.6460	**.6720**
$\Delta\%$		$(+2.96)$		$(+11.86)$		$(+4.02)$
MAP	.6086	**.6166**	.5574	**.5919***†	.5865	**.5903**
$\Delta\%$		$(+1.31)$		$(+6.19)$		$(+0.65)$

Table D.2: Performance of the standard novelty detection methods and their normalized variants (given the perfect relevance scenario).

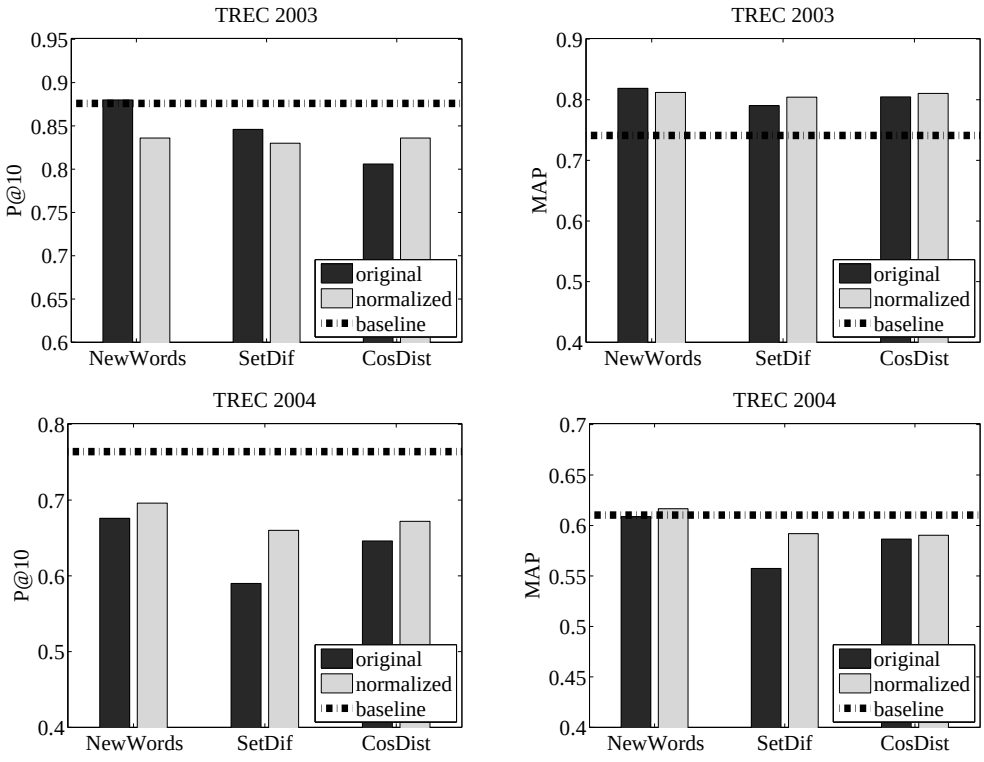
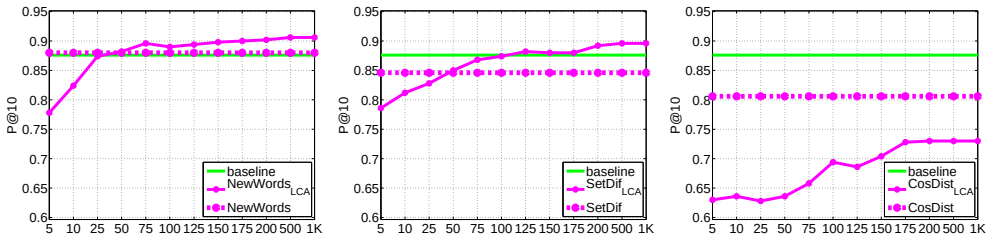


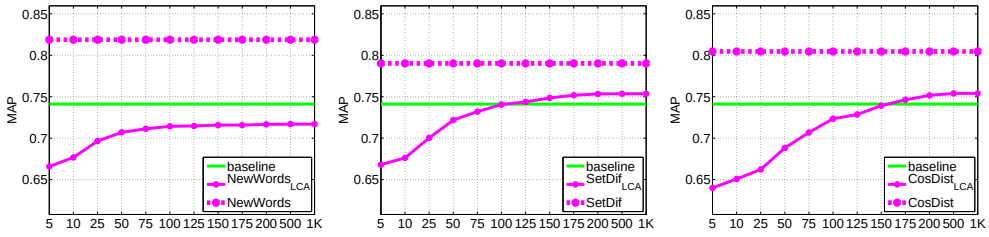
Figure D.2: Performance of the standard methods and their normalized variants (given the perfect relevance scenario).

D.3 Novelty Detection Based on Vocabulary Pruning

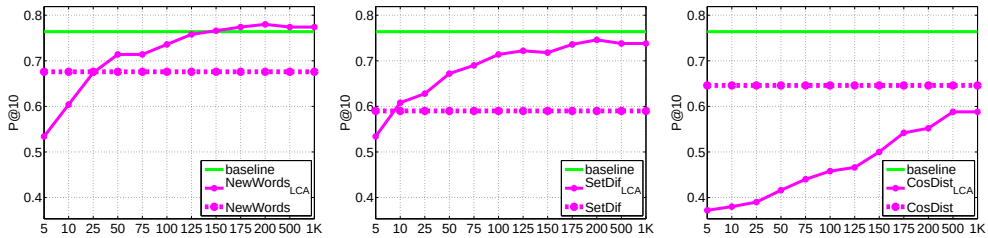
D.3.1 Local Context Analysis



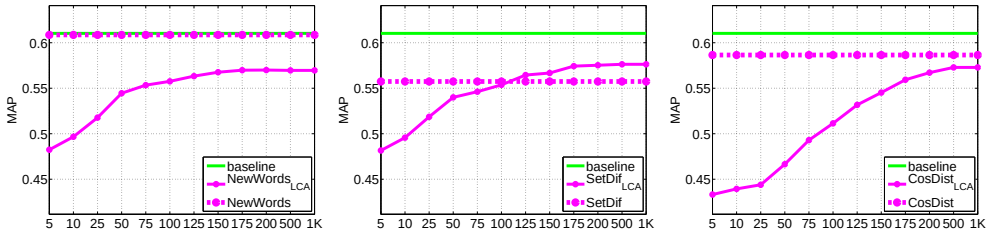
(a) P@10 TREC 2003



(b) MAP TREC 2003



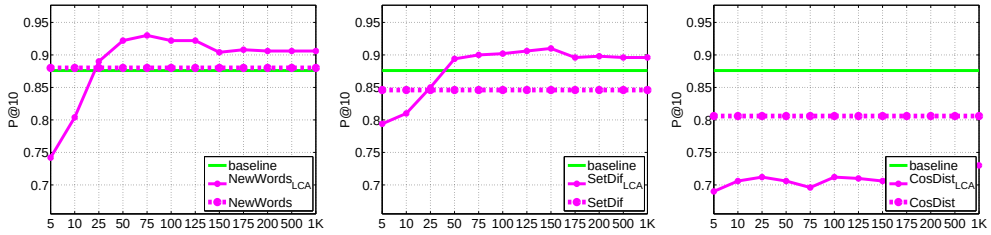
(c) P@10 TREC 2004



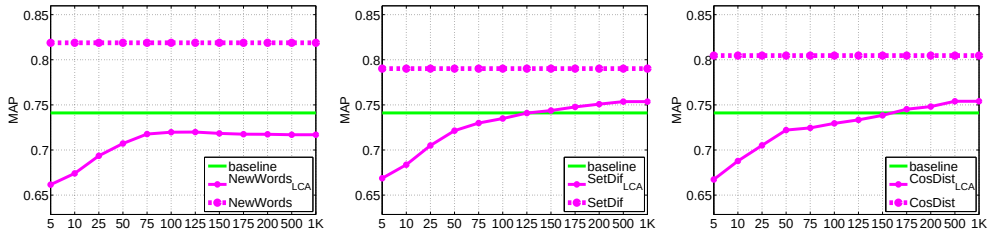
(d) MAP TREC 2004

Figure D.3: Results of state of the art novelty methods using LCA-based vocabulary pruning (given the perfect relevance scenario).

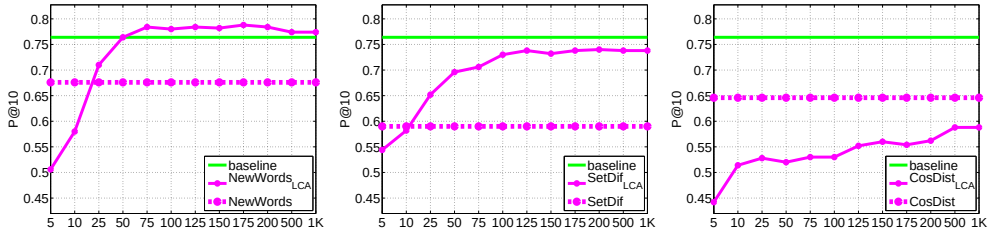
D.3.2 Divergence From Randomness



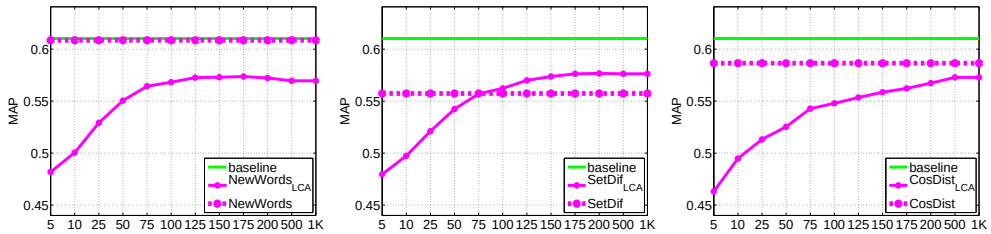
(a) P@10 TREC 2003



(b) MAP TREC 2003



(c) P@10 TREC 2004



(d) MAP TREC 2004

Figure D.4: Results of state of the art novelty methods using DFR-based vocabulary pruning (given the perfect relevance scenario).

	NewWords		SetDif		CosDist	
	basel.	**vocab.**	**basel.**	**vocab.**	**basel.**	**vocab.**
			TREC 2003			
P@10	.8800	**.9060**	.8460	**.8960***	**.8060**	.7300*
$\Delta\%$		$(+2.95)$		$(+5.91)$		(-9.43)
MAP	**.8188**	.7169*†	**.7902**	.7536*†	**.8046**	.7540*†
$\Delta\%$		(-12.45)		(-4.63)		(-6.29)
			TREC 2004			
P@10	.6760	**.7740*†**	.5900	**.7380*†**	**.6460**	.5880
$\Delta\%$		$(+14.50)$		$(+25.08)$		(-10.22)
MAP	**.6086**	.5696*†	.5574	**.5763**	**.5865**	.5729
$\Delta\%$		(-6.41)		$(+3.34)$		(-2.32)

Table D.3: Comparison of performance between standard novelty detection methods and the variant that uses vocabulary pruning (vocabulary composed of all terms in top-25 relevant sentences), given the perfect relevance scenario.

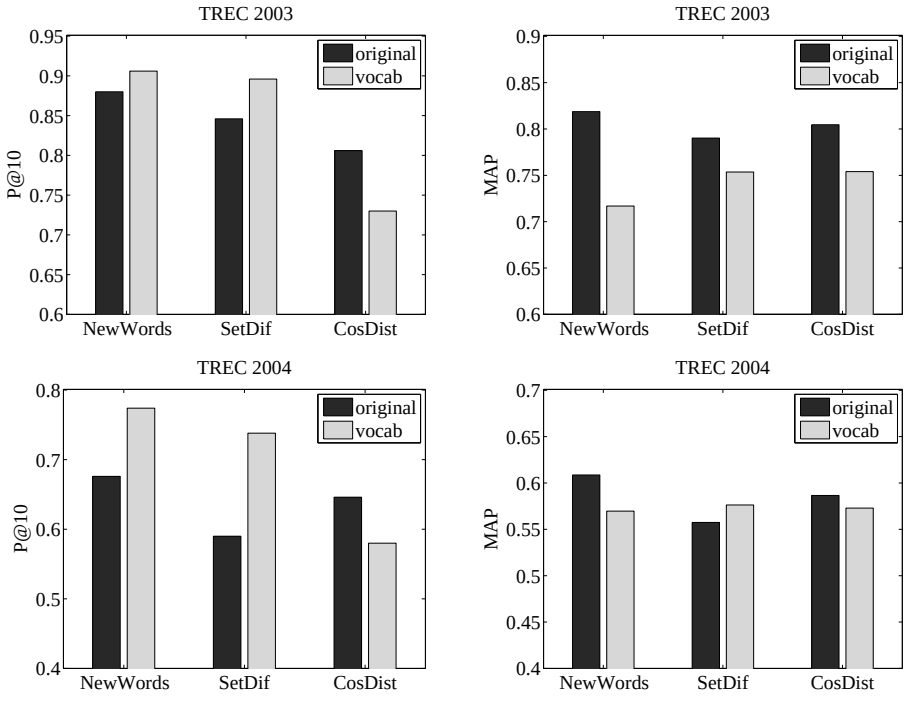
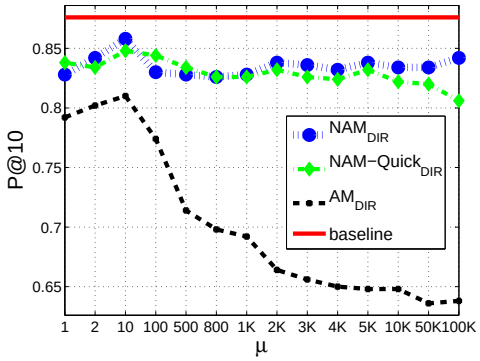
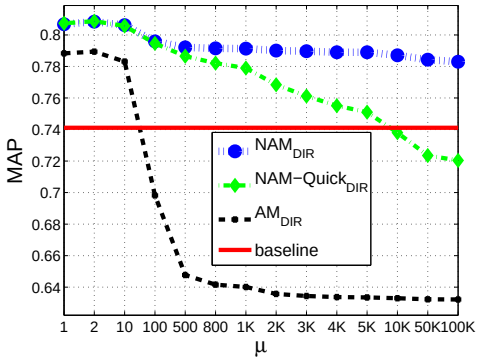


Figure D.5: Performance of the standard methods and their variants using vocabulary pruning (vocabulary composed of all terms in top 25 relevant sentences), given the perfect relevance scenario.

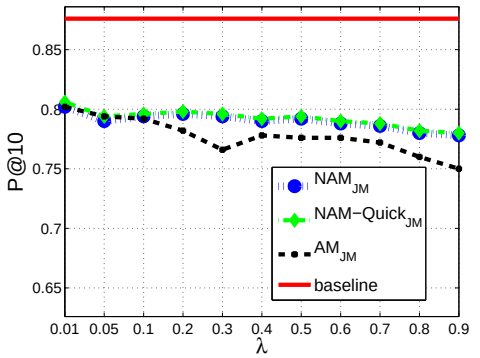
D.4 Language Modeling for the Novelty Task



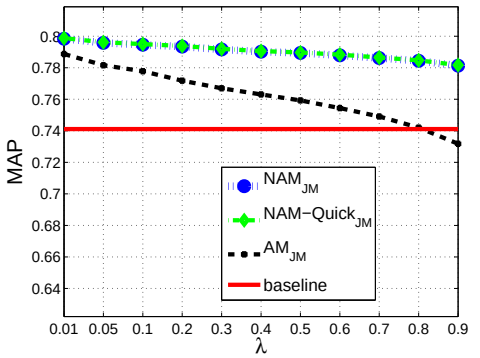
(a) P@10 TREC 2003 DIR



(b) MAP TREC 2003 DIR

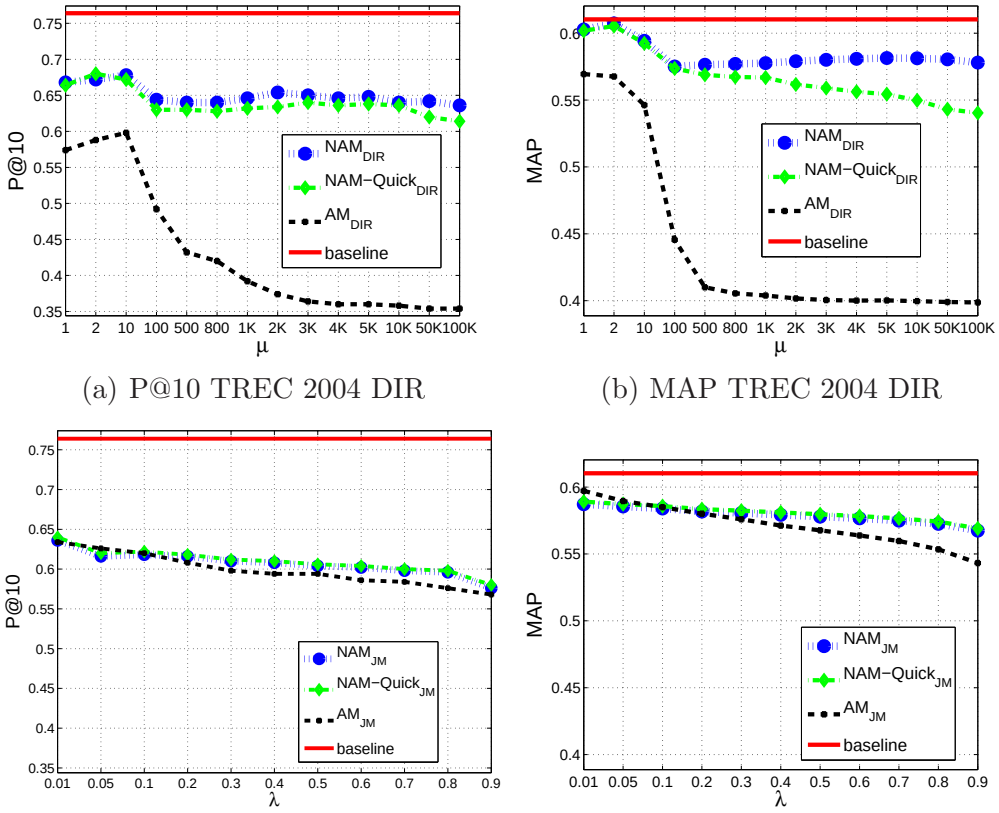


(c) P@10 TREC 2003 JM



(d) MAP TREC 2003 JM

Figure D.6: KLD-based models for novelty detection using TREC 2003 vs. perfect relevance baseline.



(a) P@10 TREC 2004 DIR (b) MAP TREC 2004 DIR

(c) P@10 TREC 2004 JM (d) MAP TREC 2004 JM

Figure D.7: KLD-based models for novelty detection using TREC 2004 vs. perfect relevance baseline.

		DIR			JM		
	baseline	**NAM**	**NAM-Quick**	**AM**	**NAM**	**NAM-Quick**	**AM**
		test: TREC 2003 (train: TREC 2004)					
P@10	**.8760**	.8580	.8340	.8020*	.8020*	.8060*	.8020*
$\Delta\%$		(-2.05)	(-4.79)	(-8.45)	(-8.45)	(-7.99)	(-8.45)
		$(\mu = 10)$	$(\mu = 2)$	$(\mu = 2)$	$(\lambda = 0.01)$	$(\lambda = 0.01)$	$(\lambda = 0.01)$
MAP	.7411	.8084*†	**.8087***†	.7883	.7984*	.7988*	.7887
$\Delta\%$		$(+9.08)$	$(+9.12)$	$(+6.37)$	$(+7.73)$	$(+7.79)$	$(+6.42)$
		$(\mu = 2)$	$(\mu = 2)$	$(\mu = 1)$	$(\lambda = 0.01)$	$(\lambda = 0.01)$	$(\lambda = 0.01)$
		test: TREC 2004 (train: TREC 2003)					
P@10	**.7640**	.6780*	.6720*	.5980*†	.6360*†	.6400*†	.6340*†
$\Delta\%$		(-11.26)	(-12.04)	(-21.73)	(-16.75)	(-20.16)	(-17.02)
		$(\mu = 10)$	$(\mu = 10)$	$(\mu = 10)$	$(\lambda = 0.01)$	$(\lambda = 0.01)$	$(\lambda = 0.01)$
MAP	**.6103**	.6073	.6054	.5676	.5872	.5891	.5972
$\Delta\%$		(-0.49)	(-0.80)	(-7.00)	(-3.78)	(-3.47)	(-2.14)
		$(\mu = 2)$	$(\mu = 2)$	$(\mu = 2)$	$(\lambda = 0.01)$	$(\lambda = 0.01)$	$(\lambda = 0.01)$

Table D.4: KLD-based models evaluated in a training-testing setting (given a perfect relevance scenario).

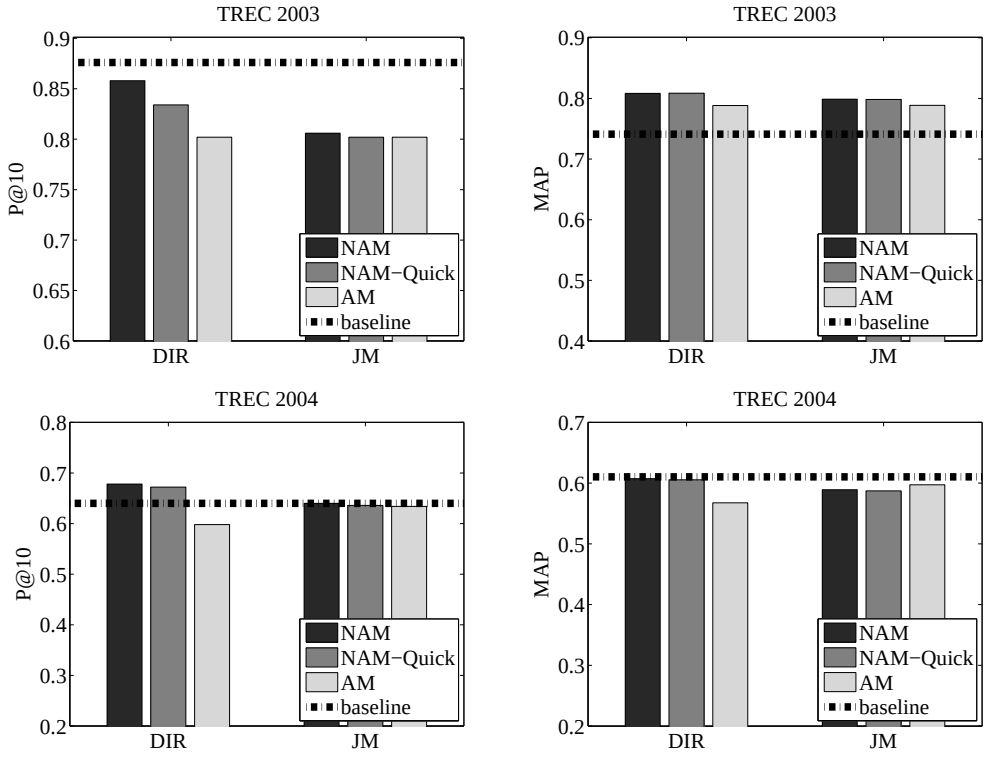


Figure D.8: Comparison among the BDOC baseline and NAM, NAM-Quick and NAM by considering DIR and JM smoothing methods (given the perfect relevance scenario).

	NAM	NAM-Quick
TREC 2003	123.79	3.70
TREC 2004	37.32	1.05

Table D.5: Time (in seconds) needed to execute a query with NAM and NAM-Quick models.

D.5 Mixture Model

	DIR			JM		
	MMEM-AM	**MMEM-NAM$_{min}$**	**MMEM-NAM$_{avg}$**	**MMEM-AM**	**MMEM-NAM$_{min}$**	**MMEM-NAM$_{avg}$**
	test: TREC 2003 (train: TREC 2004)					
P@10	μ=100000	μ=1	μ=100000	λ=0.4	λ=0.1	λ=0.1
MAP	μ=25000	μ=5	μ=1	λ=0.2	λ=0.6	λ=0.1
	test: TREC 2004 (train: TREC 2003)					
P@10	μ=1	μ=5	μ=100000	λ=0.7	λ=0.9	λ=0.1
MAP	μ=1	μ=10	μ=1	λ=0.1	λ=0.6	λ=0.1

Table D.6: Smoothing parameter values (μ/λ) for DIR and JM when building θ_R (given a perfect relevance scenario).

	baseline	DIR			JM		
		MMEM-AM	**MMEM-NAM$_{min}$**	**MMEM-NAM$_{avg}$**	**MMEM-AM**	**MMEM-NAM$_{min}$**	**MMEM-NAM$_{avg}$**
		test: TREC 2003 (train: TREC 2004)					
P@10	**.8760**	.7880*	.8500	.7840*†	.8380	.8420	.7920*
Δ%		*(−10.05)*	*(−2.97)*	*(−10.50)*	*(−4.34)*	*(−3.88)*	*(−9.59)*
MAP	.7411	.7821	**.8120***†	.7210	.8058*	.8106*†	.7214
Δ%		*(+5.53)*	*(+9.57)*	*(−2.71)*	*(+8.73)*	*(+9.38)*	*(−2.66)*
		test: TREC 2004 (train: TREC 2003)					
P@10	**.7640**	.6480*†	.6820*	.6020*†	.6560*†	.6740*	.5720*†
Δ%		*(−15.18)*	*(−10.73)*	*(−21.20)*	*(−14.14)*	*(−11.78)*	*(−25.13)*
MAP	**.6103**	.6046	.5996	.5133*†	.6067	.6006	.5122*†
Δ%		*(−0.93)*	*(−1.75)*	*(−15.89)*	*(−0.59)*	*(−1.59)*	*(−16.07)*

Table D.7: Performance of the MMEM novelty detection methods considering the perfect relevance baseline.

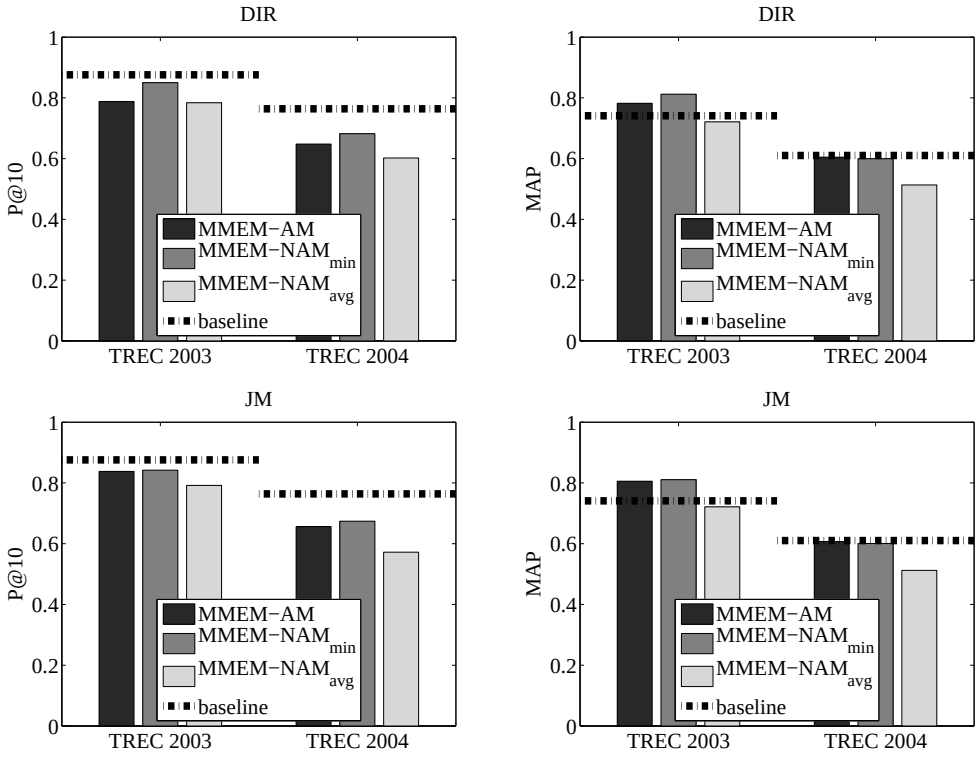


Figure D.9: Comparison between mixture model approaches and the perfect relevance baseline.

Appendix E

Resumen

Siguiendo el reglamento de los estudios de tercer ciclo de la Universidad de Santiago de Compostela, aprobado en la Junta de Gobierno el día 7 de abril de 2000 (DOG de 6 de marzo de 2001) y modificado por la Junta de Gobierno del 14 de noviembre de 2000, el Consejo de Gobierno del 22 de noviembre de 2003, del 18 de julio de 2005 (artículos 30 a 45), del 11 de noviembre de 2008 y del 14 de mayo de 2009; y, concretamente, cumpliendo las especificaciones indicadas en el capítulo 4, artículo 30, apartado 3 de dicho reglamento, mostramos a continuación un resumen en castellano de la tesis.

En esta tesis se estudia de forma exhaustiva las tareas de recuperación de información y detección de novedad. Se analizan las debilidades y puntos fuertes de los métodos que actualmente son estado del arte y, posteriormente, se proponen nuevos mecanismos capaces de recuperar frases y detectar novedad.

Las tareas de recuperación de frases y detección de novedad están relacionadas entre sí. Normalmente, se suele aplica primero un modelo de recuperación que estima de forma adecuada la relevancia de los pasajes (por ejemplo, frases) y se genera un ranking de pasajes ordenados por su relevancia. A continuación, utilizando este como entrada de un módulo de detección de novedad, se intentan filtrar los pasajes del ranking que son redundantes.

La estimación de relevancia a nivel de frase es una tarea difícil. Los métodos estándares que se utilizan para estimar relevancia se basan simplemente en la presencia de términos de la consulta en las frases. Sin embargo, las consultas suelen contener dos o tres términos únicamente y las frases tienden a ser cortas. Por lo tanto, la presencia de términos de la consulta en las frases tiende a ser muy pequeña. Para resolver este problema, en esta tesis estudiamos cómo enriquecer este proceso con información adicional: el contexto. El contexto se refiere a la información proporcionada por las frases

anexas a una dada, o bien por el documento donde se encuentra la frase. Ese contexto reduce la ambigüedad y proporciona información adicional no incluida en la propia frase. Además, es importante estimar cómo de importante o central es una frase en un documento. Estos dos componentes, el contexto y la centralidad de las frases, se estudian en esta tesis siguiendo un marco formal basado en Modelos de Lenguaje Estadísticos. Con respecto a esto, demostramos que estos componentes mejoran los métodos actuales de recuperación de frases.

En esta tesis trabajamos con colecciones de frases que han sido extraídas de noticias. Las noticias no sólo explican hechos ocurridos sino que también expresan opiniones que la gente tiene acerca de un evento concreto o un tópico. Por lo tanto, una correcta estimación de qué pasajes expresan una opinión pueden ayudar a mejorar la estimación de relevancia de las frases. Además, proponemos alternativas simples y empíricas para incorporar características independientes de la consulta en modelos de recuperación de frases. Se demuestra que la incorporación de opiniones para estimar la relevancia es un factor importante que hace que los métodos de recuperación de frases sean más eficaces. A lo largo de nuestro estudio, analizamos también características independientes de la consulta basadas en la longitud de la frase y los nombres de entidades.

La combinación de la aproximación basada en el contexto con la incorporación de características basadas en opiniones es directa. Por tanto, se estudia cómo combinar estas dos aproximaciones y el impacto de dicha combinación. Se demuestra que los modelos basados en contexto implícitamente promueven frases con opiniones y, por tanto, las características basadas en opiniones no ayudan a mejorar los métodos basados en contexto.

La segunda parte de esta tesis está dedicada a la detección de novedad a nivel de frase. Dado que la tarea de detección de novedad depende de un ranking de entrada de frases recuperadas, se consideran aquí dos aproximaciones: a) la aproximación de relevancia perfecta, que considera un ranking donde todas las frases son relevantes (esta es una aproximación ideal); y b) la aproximación de relevancia no perfecta, que consiste en aplicar primero un método de recuperación de frases (y, por lo tanto, el ranking puede contener frases que no sean relevantes).

En primer lugar, estudiamos qué líneas base funcionan mejor y, a continuación, proponemos diversas variaciones. Uno de los mecanismos que se proponen se basa en el uso de un vocabulario reducido orientado a la consulta. Demostramos que considerar sólo los términos pertenecientes a las frases ubicadas en las posiciones más altas en el ranking original ayuda a guiar la estimación de novedad. La aplicación de Modelos de Lenguaje para la detección de novedad es otro de los desafíos a los que se hace frente en esta tesis.

Aplicamos distintos modelos de suavización (Dirichlet y Jelinek-Mercer) en el contexto de mecanismos alternativos para detectar novedad (Modelos Agregado y No Agregado). Además, probamos un mecanismo basado en modelos de mezcla que utiliza el algoritmo de Maximización de la Esperanza (EM) para obtener de forma automática el grado de novedad de una frase.

En la última parte de este trabajo se demuestra que la mayoría de los métodos de detección de novedad implican una reordenación considerable del ranking inicial. Sin embargo, mostramos que las frases en las posiciones altas del ranking de entrada son normalmente noveles y una reordenación es, en este caso, perjudicial. Por lo tanto, se proponen distintos mecanismos que determinen el umbral de posición donde la detección de novedad debería iniciarse. Con respecto a esto, se consideran aproximaciones independientes de la consulta (una posición fija para todas las consultas) y aproximaciones dependiente de la consulta (basadas en agrupamiento y nivel de novedad normalizado).

Resumiendo, en esta tesis identificamos las limitaciones importantes existentes en los métodos actuales de recuperación de frases y detección de novedad y proponemos nuevos y eficaces métodos alternativos destinados a resolver estas tareas.

E.1 Recuperación de Frases y Estimación de Novedad

La recuperación de información se basa en la representación, almacenamiento, organización y acceso a elementos de información [BYRN99]. También puede ser definida como la rama de la informática que se basa en buscar material de naturaleza no estructurada en colecciones grandes (normalmente almacenadas en ordenadores) para satisfacer una necesidad de información [MRS08]. Aunque el concepto de recuperación de información es muy cercano al de búsqueda de información, la primera de las definiciones indica que la recuperación de información es una tarea más completa que incluye, además, la estructuración de información, su organización y el almacenamiento (la eficiencia y eficacia son, por tanto, dos componentes de especial importancia). Sistemas de recuperación de información ampliamente conocidos son los motores de búsqueda web (por ejemplo, Google, Yahoo!, Bing, etc.), donde los usuarios expresan su necesidad de información a travs de consultas textuales y a partir de las cuales el sistema proporciona un listado de enlaces (links) a documentos web.

La tecnología de recuperación de información está presente en muchos

ámbitos, tales como en ordenadores personales (por ejemplo, la búsqueda en el escritorio), empresas (búsqueda empresarial), etc. Los sistemas de recuperación de información tratan, normalmente, información textual. Sin embargo, otros formatos de información, como imágenes, audio y vídeo, pueden ser tratados por aplicaciones específicas de recuperación. Dado que la búsqueda de información textual es el escenario más común, en la bibliografía se hace referencia a recuperación de información y recuperación de documentos indistintamente. Sin embargo, se debe tener en cuenta que no son completamente sinónimos.

La recuperación de documentos consiste en recuperar documentos, o elementos textuales de información de un conjunto de documentos, que satisfacen una necesidad de información. La base de documentos puede encontrarse en un único ordenador (si la colección es relativamente pequeña), o distribuida en múltiples ordenadores. Por otro lado, una necesidad de información se suele expresar como una consulta de usuario. Una consulta es una secuencia de términos que describen la necesidad del usuario. Normalmente, una necesidad de información puede tener distintas consultas candidatas, aunque una consulta también podría expresar distintas necesidades de información (si no está completamente especificada o resulta ambigua). Por otro lado, dada una base de documentos y una consulta de usuario, un sistema de recuperación de documentos proporciona un ranking de documentos estimados como relevantes para la necesidad del usuario, ordenados en orden decreciente según su relevancia estimada. Este es un proceso complicado porque, normalmente, a los usuarios les es difícil traducir su necesidad de información a consultas que sean eficaces. Además, las consultas cortas son más comunes que las largas, dado que los usuarios son reacios a escribir más de dos o tres términos para una consulta (esto ocurre, especialmente, en entornos como la web [SMHM99]). De esta forma, es difícil saber de forma precisa la necesidad del usuario y, por consiguiente, identificar los documentos relevantes no es una tarea fácil para un sistema de recuperación.

La recuperación de documentos se basa en la noción de *relevancia*. Este concepto es generalmente impreciso y depende de la situación o contexto de la tarea de recuperación. Por ejemplo, la consulta *Torre Eiffel* podría expresar distintas necesidades en situaciones diferentes, tales como a) cuando una persona está planeando en ir a París y quiere saber la ubicación y precio de la entrada del monumento, o b) cuando la misma persona está en la Torre Eiffel y quiere conocer su historia. En este caso, la noción de relevancia es dependiente de la ubicación de la persona que está formulando la consulta, pero la relevancia también puede estar influenciada por otras características contextuales como la estación del año, el tiempo, la hora, el estado de ánimo, etc.

Los sistemas de recuperación de documentos normalmente tienen en cuenta dos suposiciones para simplificar sus algoritmos de recuperación: la *suposición de topicalidad de relevancia* y la *suposición de independencia de relevancia* [Zha02]. La primera indica que la relevancia de los documentos se puede medir considerando alguna forma de solapamiento entre los términos de la consulta y los términos del documento. Pero la topicalidad no es el único aspecto importante a considerar cuando se mide la relevancia de un documento. De hecho, la necesidad de ir más allá de la topicalidad (es decir, considerar información adicional proporcionada por el contexto u otras características) ha sido reconocida en la bibliografía como un aspecto importante a tener en cuenta [Sar70, Fro94]. La suposición de independencia indica que la relevancia de un documento es independiente de otros documentos. Según esto, el ranking producido por un sistema de recuperación de documentos podría contener, en las posiciones más altas, documentos que son muy similares entre sí (casi-duplicados) o idénticos (duplicados). Sin embargo, **normalmente la información redundante no es deseable**. A menudo, los usuarios están más interesados en encontrar nueva información (documentes noveles) y son menos tolerantes a obtener información que ya han visto y que, por tanto, ya conocen [Har02]. Esto significa que el sistema de recuperación debe considerar el conjunto de documentos que el usuario ha visto para estimar la relevancia de un documento. De hecho, Goffman [Gof64] enfatizó que la relevancia de un documento depende de los documentos previamente recuperados. Para resolver este problema se debe aplicar algún mecanismo de detección de novedad. Dado un conjunto ordenado de documentos (por ejemplo, los documentos estimados como relevantes por un sistema de recuperación de documentos), la detección de novedad consiste en filtrar documentos en el ranking que proporcionen información redundante, conservando sólo aquéllos que son noveles. Formalmente, Li y Croft [LC08] afirmaron que "la novedad o nueva información significa nuevas respuestas a preguntas potenciales que representan una petición de un usuario o una necesidad de información". Esta definición incluye dos aspectos: por un lado, una necesidad de usuario se podría expresar por una o más preguntas o requisitos y, por otro lado, la información novel se obtiene detectando aquellos documentos que incluyen respuestas no vistas previamente. Dependiendo del tipo de documentos noveles que le interesen a los usuarios se pueden dar dos casos: a) los usuarios podrían querer seguir buscando documentos relacionados con un tópico previamente encontrado como novel (novedad directa), o b) los usuarios podrían estar interesados en buscar documentos que no contienen información vista con anterioridad (novedad indirecta) [XY08]. La novedad directa está más bien orientada a sistemas de recuperación de información interactiva, que dependen de la interacción entre el usuario y el

sistema (el usuario marca un subtópico como novel y pide más información relacionada con ese subtópico). La novedad indirecta está más bien enfocada a satisfacer la necesidad original del usuario. En esta tesis únicamente se considera la novedad indirecta.

La detección de novedad es útil en la recuperación de documentos. Un método adecuado debe proporcionar una correcta combinación de documentos relevantes en las posiciones más altas en el ranking [WZ09]. De hecho, en un entorno real como el web, los usuarios no suelen mirar más allá de los primeros documentos en el ranking. Chen y Karger [CK06] afirmaron que intentar recuperar muchos documentos relevantes puede reducir las oportunidades de encontrar algún documento relevante, debido a la falta de diversidad.

Algunos estudios intentaron integrar novedad con topicalidad introduciendo el concepto de redundancia como el opuesto a novedad. Definieron redundancia como la cantidad de información relevante en un documento que está cubierta por documentos relevantes mostrados previamente [XY08, AWB03, ZCL03, ZCM02]. Carbonell y Goldstein [CG98] intentaron combinar la topicalidad y la novedad (como una característica independiente a la topicalidad) para estimar la relevancia de documentos. Sin embargo, muchos autores afirman que relevancia y redundancia se deben modelar de forma explícita y separada [ZCM02].

El resultado de un sistema de detección de novedad es un conjunto ordenado de documentos que son tanto relevantes como noveles. Es importante destacar que, dado que el sistema de detección de novedad se basa en un ranking de relevancia de entrada, la eficacia de la detección de novedad depende, de alguna forma, del propio ranking de relevancia.

La detección de novedad conforma un módulo importante en muchas aplicaciones potenciales en otras áreas de recuperación de información: respuesta automática a preguntas, generación automática de resúmenes, filtrado adaptativo de documentos y extracción de subtópicos. En sistemas de respuesta automática a preguntas, la consulta es una pregunta y la respuesta es un conjunto reducido de palabras que pretenden responder únicamente a lo que se pregunta. Estos sistemas buscan, por tanto, una respuesta breve y única. En este caso, los sistemas de detección de novedad son útiles porque procesan las frases que son candidatas para una pregunta dada y filtran información redundante. Los sistemas de generación automtica de resúmenes extraen un conjunto de frases que resumen brevemente un documento o conjunto de documentos. La detección de novedad es un módulo útil para esos sistemas porque las frases redundantes no se deben considerar dentro del resumen. Los sistemas de filtrado adaptativo recuperan documentos (o frases) que son relevantes para un perfil de usuario y no contienen información redundante con

respecto a documentos (o frases) previos. La detección de novedad estima si un documento (o frase) es novel con respecto a la información ya vista. Los sistemas de extracción de subtópicos extraen todos los subtópicos posibles de una consulta. Dado un texto o un pequeño trozo de información, un mecanismo de detección de novedad detecta si el texto cubre un subtópico ya tratado anteriormente, e incluso podría detectar nuevos subtópicos de carácter más genérico.

La detección de novedad también está relacionada con el concepto de diversidad. Una misma consulta puede tener más de una única interpretación, y cada interpretación puede englobar muchos subtópicos distintos [ZCL03]. Por ejemplo, dada la consulta *banco* podemos estar refiriéndonos a una entidad bancaria, a un asiento, a un conjunto de peces, etc. Además, cada una de esas interpretaciones puede englobar distintos subtópicos o facetas [CC09]. Por ejemplo, en el ejemplo de antes, dada la interpretación de entidad bancaria, el usuario puede estar interesado en saber qué es una entidad bancaria, dónde están las entidades bancarias más próximas a su domicilio, cómo se gestiona una entidad bancaria, etc. Lo deseable es que, si el sistema no es capaz de conocer la interpretación exacta o subtópico en la cual el usuario está interesado, proporcione respuestas a cada una de las posibles interpretaciones/subtópicos para la consulta. Por lo tanto, los documentos relevantes que engloban distintas interpretaciones/subtópicos se deben mostrar en posiciones más elevadas en el ranking para que la respuesta a la necesidad del usuario se proporcione lo antes posible. Nótese que, en este ámbito, la utilidad de un documento es claramente dependiente del resto de documentos en el ranking.

Los conceptos de novedad y diversidad están relacionados pero non son idénticos. Por un lado, dados dos documentos que engloban distintos subtópicos o facetas, es posible que ambos contengan información repetida. Un sistema basado en diversidad probablemente muestre ambos documentos en posiciones altas en el ranking. Sin embargo, un sistema de detección de novedad podría considerarlo redundante porque contienen información que se solapa en ambos documentos. Por otro lado, dado dos documentos, podrían clasificarse como noveles pero, aún así, podrían abarcar el mismo subtópico. Esta diferencia entre novedad y diversidad se discute más ampliamente en [XY08].

En esta tesis adoptamos la tarea de detección de novedad según su definición en las conferencias del TREC para detección de novedad (TREC Novelty Tracks) en los años 2002, 2003 y 2004 [Har02, SH03, Sob04]. En éstas, se ha dividido la tarea de novedad en dos subtareas principales: la subtarea de recuperación de frases, que consiste en, dado un conjunto de consultas y un conjunto de documentos relevantes para cada consulta, producir un conjunto

ordenado de frases; y una subtarea de detección de novedad, consistente en filtrar frases redundantes a partir de ese ranking. Considerar la detección de novedad a nivel de documento puede ser problemático porque casi todos los documentos contienen algo nuevo, particularmente cuando el dominio es el de las noticias[1] [SH05]. Para aliviar este problema, la tarea de novedad se definió a nivel de frase. Las frases son pequeños elementos de información con una estructura semántica y léxica que, a diferencia de los documentos, se caracterizan por proporcionar una pequeña idea o concepto de forma concisa. Por lo tanto, considerando frases como piezas de información es una forma natural de estudiar la detección de novedad.

La tarea de recuperación de frases consiste en buscar frases relevantes a partir de una base de documentos, dada una consulta. Esta tarea es muy útil en un amplio rango de aplicaciones de recuperación de información, tales como generación automática de resúmenes, detección de novedad, sistemas de respuesta automática y minería de opiniones. La recuperación de frases es un problema desafiante que últimamente ha llamado la atención a los investigadores del área [AWB03, WJR05, Mur06, LF07, Los08]. La inmensa cantidad de métodos de recuperación de información propuestos en la bibliografía son adaptaciones directas de modelos estándar de recuperación (como el tf-idf, BM25, Modelos de Lenguaje, etc.), donde la frase es la unidad de recuperación en lugar del documento. En estos modelos de recuperación de frases, la estimación de relevancia se basa únicamente en la presencia de términos de la consulta en las frases.

En esta tesis se proponen y estudian distintos métodos eficaces de recuperación de frases relevantes y noveles. Por un lado, se definen, se implementan y se evalúan distintas aproximaciones para resolver el problema de recuperación de frases. Esto incluye una comparación exhaustiva entre las medidas que son actualmente estado del arte. Primero, introducimos características independientes de la consulta que ayudan a estimar la relevancia de las frases. En este estudio se consideran características basadas en la presencia de nombres de entidades, la presencia de opiniones y la longitud de las frases. Cabe destacar el análisis y empleo exitoso de información basada en opiniones para recuperación de frases, que es una contribución novel en este área. Las características independientes de la consulta que se proponen ayudan a mejorar los métodos estado del arte en la recuperación de frases, sin apenas requerir costes computaciones adicionales. Por otro lado, consideramos que las frases no son piezas aisladas de información, es decir, normalmente dependen de un contexto. Este contexto normalmente

[1]Las colecciones de la TREC Novelty Track contienen documentos que son noticias extraídas de distintas fuentes de información

viene de las frases más próximas o del documento donde se encuentra la frase. Para modelar este contexto en una configuración estándar de recuperación de frases, se propone una aproximación formal, basada en Modelos de Lenguaje Estadísticos.

La segunda parte de esta tesis se dedica al estudio de detección de novedad dado un ranking de frases. Para ello, se hace un análisis en profundidad de los métodos de detección de novedad estándares y se diseñan nuevos mecanismos eficaces dados dos enfoques distintos: un enfoque de relevancia perfecta, donde se parte de un ranking de frases que han sido juzgadas como relevantes por los asesores; y un enfoque de relevancia no perfecta, donde se utiliza un ranking de frases estimadas como relevantes proporcionadas por un mecanismo de recuperación de frases. Primero, se analiza el rendimiento de los mecanismos de detección de novedad que son actualmente estado del arte y se proponen variantes a partir de esos métodos que consisten en aplicar, por ejemplo, un vocabulario limitado que enfoque el proceso de novedad únicamente en frases relacionadas con el tópico en cuestión. Además, se proponen normalizaciones noveles basadas en longitud para métodos actuales de detección de novedad. Luego, se proponen aproximaciones más formales, basadas en Modelos de Lenguaje Estadísticos, que modelan las frases como distribuciones de probabilidad y estiman la novedad como la divergencia entre esas distribuciones. A lo largo de ese estudio, también se analiza un modelo que mezcla dos modelos para detectar novedad, estimando automáticamente los parámetros (usando el algoritmo de Maximización de la Esperanza). Finalmente, se demuestra que el escenario de relevancia perfecta es más difícil de mejorar que el de relevancia no perfecta. Por lo tanto, non centramos en el caso de relevancia perfecta y proponemos nuevos mecanismos de detección de novedad basados en la congelación de las frases situadas en las posiciones más altas en el ranking.

E.2 Contribuciones

En esta tesis se lleva a cabo un análisis completo de la TREC Novelty Track, y se analiza en profundidad los problemas presentados en este ámbito.

Haciendo referencia a la recuperación de frases, las principales contribuciones son:

- Estudio comparativo del rendimiento de distintos modelos de recuperación de frases estándares, tales como tfisf, BM25 y métodos basados en Modelos de Lenguaje.

- Propuesta de características independientes de la consulta que son noveles para la recuperación de frases: características basadas en opiniones, características basadas en nombres de entidades y características basadas en la longitud de las frases.

- Aplicación fructífera de una metodología formal para incluir características independientes de la consulta en modelos existentes de recuperación de frases. Incorporando estas características en modelos de recuperación estándares se obtienen mejoras de rendimiento significativas. En particular, el efecto de características basadas en opiniones mejora mucho rendimiento de los modelos estándares.

- Estudio cuidadoso de recuperación de frases en el marco de Modelos de Lenguaje.

- Incorporación del contexto local (documento y frases anexas) en métodos basados en Modelos de Lenguaje. Esto permite desarrollar aproximaciones noveles y formales que son capaces de mejorar los métodos estados del arte.

- Incorporación de importancia de la frase en modelos de recuperación de frases siguiendo una aproximación basada en Modelos de Lenguaje. La inclusión de la importancia de la frase en modelos de recuperación conlleva mejores sustanciales en rendimiento.

- Estudio de la combinación del contexto e información basada en opiniones para estimar la relevancia de las frases.

Las principales contribuciones de esta tesis en el ámbito de la detección de novedad son:

- Estudio de detección de novedad en distintos escenarios: escenario de relevancia perfecta y no perfecta.

- Evaluación de los métodos de detección de novedad que son estado del arte en la actualidad y comparación con una línea base competitiva.

- Estudio del impacto de uso de un vocabulario limitado en métodos estándares de detección de novedad. Para obtener el vocabulario se consideraron dos mecanismos distintos: Análisis de Contexto Local (LCA) y Divergencia Desde la Aleatoriedad (DFR). También mostramos las condiciones que hacen que esta variante mejore los métodos originales.

- Evaluación de métodos formales en el contexto de Modelos de Lenguaje para resolver el problema de detección de novedad: Modelo Agregado (AM) y No-Agregado (NAM), que emplean la Divergencia Kullback-Leibler (KLD). AM considera el conjunto de frases vistas anteriormente como un todo y NAM hace comparaciones entre pares, es decir, entre una frase y cada una de las vistas anteriormente. En la bibliografía no se había hecho un estudio comparativo exhaustivo de este tipo.

- Propuesta de una variante eficaz y eficiente del modelo NAM: NAM-Quick. Este modelo es similar a NAM pero, en lugar de usar KLD, utiliza una versión modificada de KLD. Esta variante funciona al menos tan eficazmente como su versión original, pero es mucho más eficiente.

- Aplicación de un modelo de mezcla que combina un modelo de fondo (que contiene los términos del vocabulario), un modelo de referencia y un modelo para la frase para detectar novedad. Se utiliza el algoritmo de Maximización de la Esperanza (EM) para estimar automáticamente los parámetros.

A lo largo de este estudio, todas las direcciones tomadas revelan que la tarea de detección de novedad es una tarea desafiante donde es muy difícil mejorar la líneas base, que resulta ser una aproximación muy sencilla. Se proponen, además, variantes de los métodos estándares para mejorar su eficacia:

- Métodos basados en congelar las frases en posiciones más altas en el ranking y reordenar las restantes utilizando un mecanismo de detección de novedad estándar. Para ello, se emplean dos aproximaciones: una primera aproximación, basada en un umbral independiente de la consulta (donde se fija el mismo umbral para todas las consultas) y un umbral dependiente de la consulta, donde se consideran aproximaciones basadas en agrupamiento o en el grado de novedad de las frases. Se demuestra que es mejor congelar las primeras posiciones y empezar a detectar novedad en posiciones bajas del ranking. Esta es una contribución novel para la comunidad de recuperación de información en este área.

Bibliography

[AFL10] Leif Azzopardi, Ronald T. Fernández, and David E. Losada. Improving sentence retrieval with an importance prior. In *Proceedings of the* 33^{rd} *ACM International Conference on Research and Development in Information Retrieval (SIGIR 2010)*, pages 779–780, Geneva, Switzerland, 2010. ACM.

[AJR02] Gianni Amati, Cornelis Joost, and Van Rijsbergen. Probabilistic models for information retrieval based on Divergence from Randomness. *ACM Transactions on Information Systems*, 20(4):357–389, 2002.

[Ama03] Gianni Amati. *Probabilistic Models for Information Retrieval based on Divergence from Randomness*. PhD thesis, Department of Computer Science. University of Glasgow, 2003.

[AWB03] James Allan, Courtney Wade, and Alvaro Bolivar. Retrieval and novelty detection at the sentence level. In *Proceedings of the* 26^{th} *ACM International Conference on Research and Development in Information Retrieval (SIGIR 2003)*, pages 314–321, Toronto, Canada, 2003. ACM.

[BAdR09] Krisztian Balog, Leif Azzopardi, and Maarten de Rijke. A Language Modeling framework for expert finding. *Information Processing and Management*, 45(1):1–19, 2009.

[BPLF11] Rafael Berlanga, Aurora Pons, David E. Losada, and Ronald T. Fernández. *Recuperación de Información: un enfoque práctico y multidisciplinar*, chapter Técnicas avanzadas de RI. II. 2011.

[BYRN99] Ricardo Baeza-Yates and Berthier Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1999.

[BZ05] Yaniv Bernstein and Justin Zobel. Redundant documents and search effectiveness. In *Proceedings of the* 14^{th} *International Conference on Information and Knowledge Management (CIKM 2005)*, pages 736–743, Bremen, Germany, 2005. ACM.

[CC09] Ben Carterette and Praveen Chandar. Probabilistic models of novel document rankings for faceted topic retrieval. In *Proceedings of the* 18^{th} *International Conference on Information and Knowledge Management (CIKM 2009)*, pages 1287–1296, Hong Kong, China, November 2009. ACM.

[CG98] Jaime Carbonell and Jade Goldstein. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the* 21^{st} *ACM International Conference on Research and Development in Information Retrieval (SIGIR 1998)*, pages 335–336, Melbourne, Australia, 1998. ACM.

[CK06] Harr Chen and David R. Karger. Less is more: Probabilistic models for retrieving fewer relevant documents. In *Proceedings of the* 29^{th} *ACM International Conference on Research and Development in Information Retrieval (SIGIR 2006)*, pages 429–436, Seattle, USA, August 2006. ACM.

[CL03] W. Bruce Croft and John Lafferty. *Language Modeling for Information Retrieval*. Kluwer Academic Publishers, Norwell, MA, USA, 2003.

[CRZT05] Nick Craswell, Stephen E. Robertson, Hugo Zaragoza, and Michael J. Taylor. Relevance weighting for query independent evidence. In *Proceedings of the* 28^{th} *ACM International Conference on Research and Development in Information Retrieval (SIGIR 2005)*, pages 416–423, Salvador, Brazil, 2005. ACM.

[DLR77] Arthur P. Dempster, Nan M. Laird, and Donald B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977.

[Fer07] Ronald T. Fernández. The effect of smoothing in language models for novelty detection. In *Proceedings of Future Directions in Information Access (FDIA 2007)*, pages 11–16, Glasgow, UK, 2007.

[FGM05] Jenny Rose Finkel, Trond Grenager, and Christopher Manning. Incorporating non-local information into information extraction systems by Gibbs sampling. In *Proceedings of the* 43^{rd} *Annual Meeting on Association for Computational Linguistics (ICL 2005)*, pages 363–370, Ann Arbor, USA, 2005. Association for Computational Linguistics.

[FL07] Ronald T. Fernández and David E. Losada. Novelty detection using Local Context Analysis. In *Proceedings of the* 30^{th} *ACM International Conference on Research and Development in Information Retrieval (SIGIR 2007)*, pages 725–726, Amsterdam, The Nederlands, 2007. ACM.

[FL08] Ronald T. Fernández and David E. Losada. Novelty as a form of contextual re-ranking: Efficient KLD models and mixture models. In *Proceedings of* 2^{nd} *Information Interaction in Context (IIiX 2008)*, pages 27–34, London, UK, 2008.

[FL09] Ronald T. Fernández and David E. Losada. Using opinion-based features to boost sentence retrieval. In *Proceedings of the ACM* 18^th *Conference on Information and Knowledge Management (CIKM 2009)*, pages 1617–1620, Hong Kong, China, 2009. ACM.

[FLA10] Ronald T. Fernández, David E. Losada, and Leif A. Azzopardi. Extending the Language Modeling framework for sentence retrieval to include local context. *Information Retrieval*, pages 1–35, 2010.

[FPLB10] Ronald T. Fernández, Javier Parapar, David E. Losada, and Álvaro Barreiro. Where to start filtering redundancy? a cluster-based approach. In *Proceedings of the* 33^{rd} *ACM International Conference on Research and Development in Information Retrieval (SIGIR 2010)*, pages 735–736, Geneva, Switzerland, 2010. ACM.

[Fro94] Thomas J. Froehlich. Relevance reconsidered—towards an agenda for the 21^{st} century: introduction to special topic issue on relevance research. *Journal of the American Society for Information Science*, 45(3):124–134, 1994.

[Gof64] William Goffman. On relevance as a measure. *Information Storage and Retrieval*, 2(3):201–203, 1964.

[Har02] Donna Harman. Overview of the TREC 2002 Novelty Track. In *Proceedings of the* 11^{th} *Text REtrieval Conference (TREC 2002)*, pages 46–55, Gaithersburg, USA, 2002.

[Hie01] Djoerd Hiemstra. *Using Language Models for Information Retrieval*. PhD thesis, University of Twente, January 2001.

[HO07] Ben He and Iad Ounis. Combining fields for query expansion and adaptive query expansion. *Information Processing and Management*, 43(5):1294–1307, September 2007.

[JAC+04] Nasreen Abdul Jaleel, James Allan, W. Bruce Croft, Fernando Diaz, Leah S. Larkey, Xiaoyan Li, Mark D. Smucker, and Courtney Wade. UMass at TREC 2004: Novelty and HARD. In Ellen M. Voorhees and Lori P. Buckland, editors, *Proceedings of the* 13^{th} *Text REtrieval Conference (TREC 2004)*, volume Special Publication 500-261. National Institute of Standards and Technology (NIST), 2004.

[KRH04] Soo-Min Kim, Deepak Ravichandran, and Eduard Hovy. ISI novelty track system for TREC 2004. In *Proceedings of the* 13^{th} *Text Retrieval Conference (TREC 2004)*, Gaithersburg, USA, 2004.

[KSC+03] Srikanth Kallurkar, Yongmei Shi, R. Scott Cost, Charles K. Nicholas, Akshay Java, Christopher James, Sowjanya Rajavaram, Vishal Shanbhag, Sachin Bhatkar, and Drew Ogle. UMBC at TREC 12. In *Proceedings of the* 12^{th} *Text REtrieval Conference (TREC 2003)*, pages 699–706, 2003.

[KZ97] Marcin Kaszkiel and Justin Zobel. Passage retrieval revisted. In *Proceedings of the* 20^{th} *ACM International Conference on Research and Development in Information Retrieval (SIGIR 1997)*, pages 178–185, Philadelphia, USA, 1997. ACM.

[KZ01] Marcin Kaszkiel and Justin Zobel. Effective ranking with arbitrary passages. *Journal of The American Society for Informacion Science and Technology*, 52(4):344–364, 2001.

[LA08a] David E. Losada and Leif Azzopardi. An analysis on document length retrieval trends in Language Modeling smoothing. *Kluwer Academic Publishers*, 11(2):109–138, April 2008.

[LA08b] David E. Losada and Leif Azzopardi. Assessing Multi-variate Bernoulli models for information retrieval. *ACM Transactions on Information Systems (TOIS)*, 26(3):17:1–17:46, June 2008.

[LAC+02] Leah S. Larkey, James Allan, Margaret E. Connell, Alvaro Bolivar, and Courtney Wade. UMass at TREC 2002: Cross language and Novelty Tracks. In *Proceedings of the* 11^{th} *Text Retrieval Conference (TREC)*, pages 721–732, Gaithersburg, USA, 2002.

[LC02] Xiaoyong Liu and W. Bruce Croft. Passage retrieval based on Language Models. In *Proceedings of the* 11^{th} *International Conference on Information Knowledge and Management (CIKM 2002)*, pages 375–382, Virginia, USA, 2002. ACM.

[LC05] Xiaoyan Li and W. Bruce Croft. Novelty detection based on sentence level patterns. In *Proceedings of the* 14^{th} *International Conference on Information and Knowledge Management (CIKM 2005)*, pages 744–751, Bremen, Germany, 2005. ACM.

[LC08] Xiaoyan Li and W. Bruce Croft. An information pattern based approach to novelty detection. *Information Processing and Management*, 44(3):1159–1188, May 2008.

[LCA08] Kyung Soon Lee, W. Bruce Croft, and James Allan. A cluster-based resampling method for pseudo-relevance feedback. In *Proceedings of the* 31^{st} *ACM International Conference on Research and Development in Information Retrieval (SIGIR 2008)*, pages 235–242, Singapore, July 2008. ACM.

[LF07] David E. Losada and Ronald T. Fernández. Highly frequent terms and sentence retrieval. In *Proceedings of the* 14^{th} *String Processing and Information Retrieval Symposium (SPIRE 2007)*, Lecture Notes in Computer Science, pages 217–228, Santiago de Chile, Chile, 2007. Springer-Verlag.

[Li06] Xiaoyan Li. *Sentence Level Information Patterns for Novelty Detection*. PhD thesis, University of Massachusetts at Amherst, September 2006.

[LMP01] John Lafferty, Andrew McCallum, and Fernando Pereira. Conditional. random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the* 18^{th} *International Conference on Machine Learning (ICML 2001)*, pages 282–289, Williamstown, USA, 2001.

[Los08] David E. Losada. A study of statistical query expansion strategies for sentence retrieval. In *Proceedings SIGIR 2008 Workshop on Focused Retrieval (Question Answering, Passage Retrieval, Element Retrieval)*, Singapore, 2008. ACM.

[Los10] David E. Losada. Statistical query expansion for sentence retrieval and its effects on weak and strong queries. *Information Retrieval*, 13(5):485–506, 2010.

[MHPO04] Craig Macdonald, Ben He, Vassilis Plachouras, and Iad Ounis. University of Glasgow at TREC 2004: Experiments in Terabyte and Enterprise Tracks with Terrier. In *Proceedings of the* 13^{th} *Text REtrieval Conference (TREC 2005)*, Gaithersburg, USA, 2004.

[MHPO05] Craig Macdonald, Ben He, Vassilis Plachouras, and Iad Ounis. University of Glasgow at TREC 2005: Experiments in Terabyte and Enterprise Tracks with Terrier. In *Proceedings of the* 14^{th} *Text REtrieval Conference (TREC 2005)*, Gaithersburg, USA, 2005.

[MK08] Geoffrey J. McLachlan and Thriyambakam Krishnan. *The EM algorithm and extensions*. Wiley Series in Probability and Statistics, 2^{nd} edition, 2008.

[MLS99] David R. Miller, Tim Leek, and Richard M. Schwartz. A Hidden Markov Model information retrieval system. In *Proceedings of the* 22^{nd} *ACM International Conference on Research and Development in Information Retrieval(SIGIR 1999)*, pages 214–221, Berkeley, US, 1999. ACM.

[MOS07] Craig Macdonald, Iadh Ounis, and Ian Soboroff. Overview of the TREC-2007 Blog Track. In *Proceedings of the* 16^{th} *Text REtrieval Conference (TREC 2007)*, 2007.

[MOS09] Craig Macdonald, Iadh Ounis, and Ian Soboroff. Overview of the TREC-2009 Blog Track. In *Proceedings of the* 18^{th} *Text REtrieval Conference (TREC 2009)*, 2009.

[MRS08] Christopher Manning, Prabhakar Raghavan, and Hinrich Schtze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[Mur06] Vanessa G. Murdock. *Aspects of sentence retrieval*. PhD thesis, University of Massachusetts Amherst, September 2006.

[OdRM+06] Iadh Ounis, Maarten de Rijke, Craig Macdonald, Gilad Mishne, and Ian Soboroff. Overview of the TREC-2006 Blog Track. In *Proceedings of the* 15^{th} *Text Retrieval Conference (TREC 2006)*, 2006.

[OMS08] Iadh Ounis, Craig Macdonald, and Ian Soboroff. Overview of the TREC-2008 Blog Track. In *Proceedings of the* 17^{th} *Text REtrieval Conference (TREC 2008)*, 2008.

[PC98] Jay M. Ponte and W. Bruce Croft. A Language Modeling approach to information retrieval. In *Proceedings of the* 21^{st} *ACM International Conference on Research and Development in Information Retrieval (SIGIR 1998)*, pages 275–281, Melbourne, Australia, 1998. ACM.

[PL08a] Bo Pang and Lillian Lee. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2(1-2):1–135, 2008.

[PL08b] Bo Pang and Lillian Lee. Using very simple statistics for review search: An exploration. In *Proceedings of COLING: Companion volume: posters*, pages 73–76, Manchester, UK, 2008.

[Rob05] Stephen Robertson. *In Book TREC: Experiment and Evaluation in Information Retrieval*, chapter How Okapi came to TREC, pages 287–299. Digital Libraries and Electronic Publishing. MIT Press, 2005.

[RWJ+94] Stephen E. Robertson, Steve Walker, Susan Jones, Micheline Hancock-Beaulieu, and Mike Gatford. Okapi at TREC-3. In *Proceedings of the* 3^{rd} *Text REtrieval Conference (TREC 1994)*, pages 109–126, Gaithersburg, USA, 1994.

[RZT04] Stephen E. Robertson, Hugo Zaragoza, and Michael J. Taylor. Simple BM25 extension to multiple weighted fields. In *Proceedings of the* 13^{th} *International Conference on Information and Knowledge Management (CIKM 2004)*, pages 42–49, Washington, USA, 2004. ACM.

[SA05] Mark D. Smucker and James Allan. An investigation of Dirichlet prior smoothing's performance advantage. Technical report, University of Massaghussetts, Amherst, CIIR, 2005.

[SAC07] Mark D. Smucker, James Allan, and Ben Carterette. A comparison of statistical significance tests for information retrieval evaluation. In *Proceedings of the* 16^{th} *International Conference on Information and Knowledge Management (CIKM 2007)*, pages 623–632, Lisbon, Portugal, 2007. ACM.

[Sar70] Tefko Saracevic. *Introduction to information science*, chapter The concept of "relevance" in information science: A historical review, pages 111–154. R. R Bowker, New York, 1970.

[SBMM96] Amit Singhal, Chris Buckley, Mandar Mitra, and Ar Mitra. Pivoted document length normalization. In *Proceedings of the* 19^{th} *ACM International Conference on Research and Development in Information Retrieval (SIGIR 1996)*, pages 21–29. ACM, 1996.

[SH03] Ian Soboroff and Donna Harman. Overview of the TREC 2003 Novelty Track. In *Proceedings of the* 12^{th} *Text REtrieval Conference (TREC 2003)*, Gaithersburg, USA, 2003.

[SH05] Ian Soboroff and Donna Harman. Novelty detection: the TREC experience. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing (HLT 2005)*, pages 105–112, Vancouver, Canada, 2005. Association for Computational Linguistics.

[SJCO02] Luo Si, Rong Jin, Jamie Callan, and Paul Ogilvie. A Language Modeling framework for resource selection and results merging. In *Proceedings of the ACM* 11^th *Conference on Information and Knowledge Management (CIKM 2002)*, pages 391–397, New York, USA, 2002. ACM.

[SMHM99] Craig Silverstein, Hannes Marais, Monika Henzinger, and Michael Moricz. Analysis of a very large web search engine query log. *SIGIR Forum*, 33(1):6–12, 1999.

[Sob04] Ian Soboroff. Overview of the TREC 2004 Novelty Track. In *Proceedings of the* 13^{th} *Text Retrieval Conference (TREC 2004)*, Gaithersburg, USA, 2004.

[TTC10] Flora S. Tsai, Wenyin Tang, and Kap Luk Chan. Evaluation of novelty metrics for sentence-level novelty mining. *Information Sciences*, 180(12):2359–2374, 2010.

[Voo93] Ellen M. Voorhees. Using WordNet to disambiguate word senses for text retrieval. In *Proceedings of the* 16^{th} *ACM International Conference on Research and Development in Information Retrieval (SIGIR 1993)*, pages 171–180, New York, NY, USA, 1993. ACM.

[WJR05] Ryen W. White, Joemon M. Jose, and Ian Ruthven. Using top-ranking sentences to facilitate effective information access. *American Society for Information Science and Technology*, 56(10):1113–1125, 2005.

[WR05] Janyce Wiebe and Ellen Riloff. Creating subjective and objective sentence classifiers from unannotated texts. In *Proceedings of the Conference on Computational Linguistics and Intelligent Text Processing (CICLing 2005)*, Lecture Notes in Computer Science, pages 475–486, Mexico City, Mexico, 2005. Springer-Verlag.

[WZ09] Jun Wang and Jianhan Zhu. Portfolio theory of information retrieval. In *Proceedings of the* 32^{nd} *ACM International Conference on Research and Development in Information Retrieval (SIGIR 2009)*, pages 115–122, Boston, USA, July 2009. ACM.

[XC96] Jinxi Xu and W. Bruce Croft. Query expansion using local and global document analysis. In *Proceedings of the* 19^{th} *ACM International Conference on Research and Development in Information Retrieval (SIGIR 1996)*, pages 4–11, Zurich, Switzerland, 1996. ACM.

[XC00] Jinxi Xu and W. Bruce Croft. Improving the effectiveness of information retrieval with Local Context Analysis. *ACM Transactions on Information Systems (TOIS)*, 18(1):79–112, January 2000.

[XJC08] Xiaobing Xue, Jiwoon Jeon, and W. Bruce Croft. Retrieval models for question and answer archives. In *Proceedings of the* 31^{st} *ACM International Conference on Research and Development in Information Retrieval (SIGIR 2008)*, pages 475–482, Singapore, 2008. ACM.

[XY08] Yunjie Xu and Hainan Yin. Novelty and topicality in interactive information retrieval. *American Society for Information Science and Technology*, 59(2):201–215, January 2008.

[ZCL03] ChengXiang Zhai, William W. Cohen, and John Lafferty. Beyond independent relevance: Methods and evaluation metrics for subtopic retrieval. In *Proceedings of the* 26^{th} *ACM International Conference on Research and Development in Information Retrieval (SIGIR 2003)*, pages 10–17, Toronto, Canada, 2003. ACM.

[ZCM02] Yi Zhang, Jamie Callan, and Thomas Minka. Novelty and redundancy detection in adaptive filtering. In *Proceedings of the* 25^{th} *ACM International Conference on Research and Development in Information Retrieval (SIGIR 2002)*, pages 81–88, Tampere, Finland, 2002. ACM.

[Zha02] ChengXiang Zhai. *Risk Minimization and Language Modeling in Text Retrieval*. PhD thesis, Computer Science, Canergie Mellon University, July 2002.

[ZL01] Chengxiang Zhai and John Lafferty. A study of smoothing methods for Language Models applied to ad hoc information retrieval. In *Proceedings of the* 24^{th} *ACM International Conference on Research and Development in Information Retrieval (SIGIR 2001)*, pages 334–342, New Orleans, USA, 2001. ACM.

[ZL02] Chengxiang Zhai and John Lafferty. Two-stage Language Models for information retrieval. In *Proceedings of the* 25^{th} *ACM International Conference on Research and Development in Information Retrieval(SIGIR 2002)*, pages 49–56. Kluwer Academic Publishers, 2002.

[ZL04] Chengxiang Zhai and John Lafferty. A study of smoothing methods for Language Models applied to information retrieval. *ACM Transactions on Information Systems (TOIS)*, 22(2):179–214, April 2004.

[ZLL+03] Min Zhang, Chuan Lin, Yiqun Liu, Leo Zhao, and Shaoping Ma. THUIR at TREC 2003: Novelty, robust and web. In *Proceedings of the* 12^{th} *Text REtrieval Conference (TREC 2003)*, pages 556–567, Gaithersburg, USA, 2003.

[ZZM06] Le Zhao, Min Zhang, and Shaopin Ma. The nature of novelty detection. *Information Retrieval*, 9(5):521–541, November 2006.