Proximity of Terms, Texts and Semantic Vectors in Information Retrieval

Proximity of Terms, Texts and Semantic Vectors in Information Retrieval

Proefschrift

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus prof. ir. K.C.A.M. Luyben,
voorzitter van het College voor Promoties,
in het openbaar te verdedigen op

door

Jeroen Bastiaan Pieter Vuurens

geboren te Amstelveen, Nederland.

Samenstelling promotiecommissie:

Rector Magnificus,	voorzitter
Prof. dr. ir. A. P. de Vries,	Technische Universiteit Delft

Onafhankelijke leden:

Overige leden:

Keywords: Information Retrieval, Retrieval Algorithms, Clustering, Recommender Systems

Printed by: Name Printing Company

Front & Back: Cover art by Martijn de Jongh.

ISBN 000-00-0000-000-0

An electronic version of this dissertation is available at
`http://repository.tudelft.nl/`.

Alice laughed:"There's no use trying," she said; "one can't believe impossible things."
"I daresay you haven't had much practice," said the Queen.
"When I was younger, I always did it for half an hour a day.
Why, sometimes I've believed as many as six impossible things before breakfast."

Alice in Wonderland

CONTENTS

SUMMARY

Information Retrieval (IR) is finding content of an unstructured nature with respect to an information need. A retrieval system typically uses a retrieval model to rank the available content by their estimated relevance to an information need. For decades, state-of-the-art retrieval models have used the assumption that terms appear independently in text documents. Chapter 1 of this thesis describes how the relevance likelihood of a document changes by the observed distance between co-occurring query terms in its text.

Nowadays, news is abundantly available online, allowing users to discover and follow news events. However, online news is often very redundant; most sources basing their stories on previously published works and add only limited new information. Thus, a user often ends up spending significant amount of effort re-reading the same parts of a story before finding relevant and novel information. In Chapter 2 and Chapter 3, we present a novel approach to construct an online news summary for a given topic. Salient sentences are identified by clustering the sentences in the news stream based on the relative proximity of the sentences and the temporal proximity of their publication times. To improve the coherence of a long summary that describes a news topic, we propose to automatically cluster sentences by subtopics in Chapter 4. In Chapter 5, we show how new topics can be detected in the news stream using the same clustering technique.

In real-life decision making, people are often faced with an overload of choices. A recommender system aids the user by reducing the available choices to a shortlist of items that are of interest to the user. In Chapter 6, we learn high-dimensional representations for movies that allow to effectively recommend movies based on a user's most recently rated movies.

SAMENVATTING

Information Retrieval (IR) is het vinden van ongestructureerd materiaal met betrekking tot een informatiebehoefte. Een zoeksysteem schat de relevantie van het beschikbare materiaal voor een informatiebehoefte om het daarnaar te rangschikken. Voor het schatten van de relevantie gingen toonaangevende modellen er tot voor kort van uit dat woorden onafhankelijk van elkaar voorkomen in tekst documenten. Hoofdstuk 1 van deze dissertatie beschrijft hoe de geschatte relevantie van een document afhangt van de woordafstand tussen zoekvraag termen in de tekst.

Vandaag de dag is nieuws volop op Internet beschikbaar, waarmee gebruikers gebeurtenissen in het nieuws kunnen ontdekken en volgen. Het nieuws op Internet is echter zeer redundant; de meeste artikelen zijn gebaseerd op eerder gepubliceerd werk en voegen daar maar beperkt nieuwe informatie aan toe. Als gevolg moet een gebruiker vaak grote hoeveelheden bekende informatie lezen voordat nieuwe en relevante informatie wordt gevonden. In Hoofdstuk 2 en Hoofdstuk 3 presenteren we een nieuwe aanpak voor het samenvatten van het nieuws over een gegeven onderwerp. Zinnen die nieuwswaarde bevatten worden gevonden door de zinnen uit nieuwsartikelen te clusteren op basis van de afstand tot andere zinnen en de tijd die tussen de publicaties zit. Om de leesbaarheid van een lange samenvatting over een nieuwsonderwerp te verhogen stellen we een aanpak voor die de zinnen automatisch indeelt naar subtopic (Hoofdstuk 4). In Hoofdstuk 5, tonen we hoe nieuwe gebeurtenissen kunnen worden ontdekt in nieuws artikelen met behulp van dezelfde cluster techniek.

Mensen worden voor het nemen van beslissingen regelmatig overladen met keuzemogelijkheden. Een recommender system kan alle beschikbare mogelijkheden terugbrengen tot een selectie van alternatieven die voor een gebruiker het meest interessante zijn. In Hoofdstuk 6, gebruiken we hoog-dimensionale representaties voor films om aan een gebruiker geschikte films aan te bevelen op basis van haar meest recente waardering voor films.

INTRODUCTION

Information Retrieval (IR) is finding material of an unstructured nature with respect to an information need (Manning et al., 2008). In contrast to data retrieval, which addresses the task of obtaining factual information from structured data using well defined semantics, IR algorithms are used to estimate which items are most relevant to a user need from a collection of weakly-structured items, e.g. webpages, movies. Over the last decades, IR systems have evolved from basic document retrieval engines into more profound systems that tailor the information to a user's need or interest. Considering that the Web is a very large data space, efforts are made to reduce the information overload that users may face, for instance by representing documents using text snippets and prioritizing recommended items from large assortments. The continuous improvement of IR effectiveness has driven web search engines to a level where most people consider the Internet to be a good place for getting everyday information.

Over the past decade there has been an unabated growth of online content, not just consisting of text documents, but also including the hyperlink structure between web pages, social user networks, user-generated content, sensory data, and auxiliary datasets such as geographical maps. As a research domain IR is exciting and evolving in a rapid pace. On the one hand, the increase of available data has served as a catalyst for the innovation of new or improved uses, for instance maps that integrate links to restaurants with user reviews, and online shops with recommended items that are tailored to a user's interest. On the other hand, innovations are also driven by commerce and the technology push in other domains, such as mobile phone development, speech recognition, connectivity, virtual/augmented reality, and artificial intelligence.

PROXIMITY

For machines it may be very hard to fully grasp the meaning that humans assign to objects, given that for instance human attempts to formalize english grammar are very elaborate while still considered incomplete (e.g. Biber et al. (1999) contains a description that is over 1100 pages). Box's famous aphorism that "essentially, all models are wrong, but some are useful" certainly applies to IR. However, if the task is to estimate which objects are likely to be most interesting to a user, a simple model may suffice (Halevy et al., 2009). To simplify matters, at the basis of many data science algorithms are features; aspects of objects that are indicative of its usefulness for a specific need. Features are not limited to explicit item attributes; a great deal of 'mileage' can be achieved by designing additional features which are suited to a specific problem (Manning et al., 2008).

The cluster hypothesis states that documents in the same cluster behave similarly with respect to relevance to information needs (Van Rijsbergen, 1979). Specifically in the context of a query, if we find one relevant item, the items that are most similar to this item are also likely to be relevant (Tombros, 2002). This assumption has shown to be very useful for a wide range of tasks such as document retrieval, item recommendation, and item classification. As an extension, this research looks into the proximity between terms, sentences, and semantic vectors, to learn how proximity can be used to design features that can improve the relevance estimation for different tasks.

RESEARCH GOAL

This dissertation aims to answer how proximity can be useful to estimate the relevance of items or information, for which three different aspects are addressed:

Q1: how is the proximity between terms indicative of the relevance of a document?

Q2: how is the proximity between published sentences in news articles indicative for the salience and novelty of news?

Q3: how can the proximity between semantic item vectors be used as the basis of item recommendation?

METHODOLOGIES IN IR

The challenge in IR is one of managing uncertainty and computational complexity. Turtle and Croft (1997) identify three major components for IR for which uncertainty has to be dealt with: the document (representation), the information need and the matching function. The description of the information need is considered to be imperfect for several reasons: a user may not exactly know what they are looking (Belkin et al., 1982), users tend to put minimal effort in describing their need (Jansen et al., 2000), and implicit user needs such as past preferences are not always the best predictors of future needs. Additionally, a computer system is never able to fully comprehend the value of an item to the user, hence the correctness of internal representations is difficult to validate. And even if we could show the 'correctness' of internal representations, good scores on an internal criterion do not necessarily translate into good effectiveness in an application (Manning et al., 2008). Therefore, to show that IR research is meaningful requires an evaluation of its usefulness towards a user's need.

Information Retrieval research has developed itself as a highly empirical discipline (Manning et al., 2008). We can describe the research process using the empirical cycle as an overall methodology (Figure 1). In the first step, a problem is observed and the related data analyzed, often sparking an interesting idea to study. Then with inductive reasoning a hypothesis is formed, which is implemented into a design and evaluated against state-of-the-art solutions for the task that is addressed.

To demonstrate that IR research is meaningful and that a proposed model performs superior, requires that the proposed models are evaluated over their usefulness to the end-user, for instance by measuring the effort needed from the user to complete their task using a system's output. However, in practice an evaluation of this form is often not feasible. In industry, one feasible alternative is to apply online A/B tests, but since an explicit confirmation of user satisfaction is hard to obtain, satisfaction is commonly inferred from implicit signals such as clicks, time-spend-on-page or bookmarks. A commonly used alternative in offline/academic settings was introduced in the Cranfield ex-

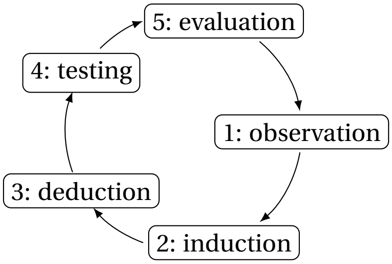


Figure 1: Empirical cycle.

periments, in which expert annotators score the relevance of the information that is retrieved for a specific information need. The amount of effort that is required by the user to satisfy their information need is then often expressed in an evaluation metric that considers the amount of irrelevant items that clutter the results or the extent to which the relevant items are ranked above the irrelevant items. Carterette and Allan (2007) stress that for IR research the scientific results need strong support in order to be accepted. In the IR research community, for a result to gain broad acceptance, it must have been tested on multiple corpora, compared to strong baselines, and shown to be statistically significant.

During this research, our personal experience is that during the first step of the empirical cycle it is most critical that a thorough understanding of the problem is obtained; typically by an extensive analysis of the problem and available data to understand which features could be helpful to address the problem. In Data Mining methodologies this is sometimes referred to as “data understanding” (Chapman et al., 2000), which comes before hypothesis forming. With regards to the evaluation, we stress that to show that the improvement over state-of-the-art baselines transfers to other domains (i.e. generalization), the evaluation should consider a *variety* of corpora and a comparison against strong baselines.

The next three subsections will introduce the three parts of this thesis, explain how the empirical cycle was applied and how the evaluation was conducted to show the significance of these studies.

TERM PROXIMITY

One of the core topics for IR research is the retrieval of documents from a corpus. A traditional setup for a document retrieval system uses an index over the words that appear in the collections’ documents, to efficiently estimate the relevance of the documents based on the words they share with a natural language query. For decades, state-of-the-art retrieval models have used the assumption that terms appear independently in text documents, which not only allows for very efficient models, but also appeared surprisingly difficult to improve over by adding term dependencies; Lavrenko (2004, p. 13) noted in his thesis:

> “No other aspect of the formalism has drawn so much criticism and failed endeavors to improve the model. It is my personal observation that almost

> every mathematically inclined graduate student in Information Retrieval attempts to formulate some sort of a non-independent model of IR within the first two or three years of his studies. The vast majority of these attempts yield no improvements and remain unpublished."

However, Manning et al. (2008) argue that especially for free text queries on the web, users prefer a document in which most or all of the query terms appear close to each other, because this is evidence that the document has text focused on their query intent. Metzler and Croft (2005) argue:

> "It is well known that dependencies exist between terms in a collection of text. For example, within a SIGIR proceedings, occurrences of certain pairs of terms are correlated, such as 'information' and 'retrieval'. The fact that either one occurs provides strong evidence that the other is also likely to occur. Unfortunately, estimating statistical models for general term dependencies is infeasible, due to data sparsity. For this reason, most retrieval models assume some form of independence exists between terms."

In 2005, Metzler and Croft show that by adding features that score the occurrence of adjacently appearing query terms and query terms that appear in close proximity in text, retrieval effectiveness is consistently improved. But rather than understanding how the distribution of words over text should be interpreted with respect to their relevance score, machine learning is used to find an optimal mixture over term independence and term dependence functions.

Part I of this thesis analyses how the distance between query terms that appear in documents can be used to improve the relevance score assigned to the documents. An oracle study is performed to assess the feasibility of term proximity models for document retrieval, by trying to improve the effectiveness of queries using weighted proximity expansions that match a combination of terms within a number of word positions. From this oracle study we learn that promoting documents that contain selected combinations of query terms within a fixed word span leads to more effective results, but also that optimal results are sometimes obtained using proximity expansions that span a distance of a hundred words or more. We analyze the distribution of distance between query terms in relevant documents vs. non-relevant documents in TREC collections, to understand how the distance between query terms in a document corresponds to the expected relevance of that document. This analysis reveals that documents are twice as likely to be relevant when they contain two query terms adjacently than when they contain two query terms separated by many other words. For intermediate distances between query terms in a document, the likelihood that a document is relevant decreases by a function that is 1 over the number of words that separate them. We hypothesize that the estimation of relevance for a document is improved by extending the relevance score of a term-independence baseline using the observed distance between cooccurring query terms.

In the evaluation, we show that this model obtains equals or better results when compared to state-of-the-art term dependency models, over an extensive range of news wire and Web collections. We also participated in the TREC 2013 Web Track with the proposed retrieval model, and obtained an average precision that was comparable to the best system which used machine learning over a range of features. In independent recent work, Roy et al. (2016) conclude that our model is more robust than other state-of-the-art term dependency models and make a comparison over two additional datasets. Additionally, they show that the model may be simplified in favor of higher efficiency.

NEWS SUMMARIZATION

Internet users are turning more frequently to online news as a replacement for traditional media sources such as newspapers or television shows. Still, discovering news events online and following them as they develop can be a difficult task. Although the Web offers a seemingly large and diverse set of information sources ranging from highly curated professional content to social media, in practice most sources base their stories on previously published works and add only limited new information. Thus, a user often ends up spending a significant amount of effort re-reading the same parts of a story before finding relevant and novel information.

In Part II of this thesis, we study the online detection and tracking of topics in news articles. To track a news topic, sentences are extracted from the stream of news articles to form a timely and concise overview of the most important information regarding that topic in the news. Additional work is presented on clustering the sentences within an obtained news summary by subtopics, and on the detection of novel topics in the news stream.

To identify when interesting news is published, our initial idea was that important news will lead to an increase in queries issued by users, and thus should be observable in the query logs of a search engine. We analyzed the Yahoo! query log for 20 hand picked topics (e.g. New York Yankees, Maria Sharapova, Emmy Awards) during an interval of 9 months, in which each topic was headlined in the news multiple times. We used a ground truth that was extracted from the topics' Wikipedia edit pages, which includes a timestamp when new information was added. For these topics, we analyzed the correlation between query bursts and the news that was added to their Wikipedia page, and were surprised to find that the majority of bursts does not correspond to news being published. The query log registers where traffic is originated from, which allowed us to trace back bursts that are not related news events. It appeared many of these bursts are the result of traffic generated by single websites such as the Yahoo! Trending Now referral system. Most referral systems suggest trending queries, but being optimized to generate traffic they often favor popular entities over recent news topics. As a result we found many bursts originating in different causes than actual news, and decided upon a new research direction.

Since query logs appeared to be a poor indicator of important news, we reformulated the problem as:

- how can we infer whether a news article contains salient news?
- how can we 'grow' a summary of sentences, maximizing coverage of the latest and most important news, while minimizing redundancy?

Our analysis of the news articles that contain the information that was put on Wikipedia, shows that important news is often published by multiple newspapers within a short timespan which between them use a great number of identical words to describe the news facts. The latter could be the result of many news papers obtaining their information from the same source, e.g. Reuters, Associated Press. Therefore, we hypothesize that salient news can be identified by multiple news papers publishing very similar information in close temporal proximity. We propose a 3-nearest neighbor clustering algorithm (3NN) that clusters information based on similarity in content and publication time, to identify salient sentences in news articles as newly formed clusters, which effectively filters the news stream. When following a news topic, the salient sentences from clusters that 'match' the targeted topic are added to a relevance model of the information seen on that topic most recently. Then a sentence is added to the summary when it contains novel information and when it ranks highly according to the relevance model amongst the sentences already in the summary.

For the evaluation, we used the 2013 and 2014 TREC Temporal Summarization Track test sets. The results show that although 3NN has a higher latency, the recall is comparable to that of top systems and the precision significantly better. We participated at the 2015 TREC Temporal Summarization Track wth the proposed model, which confirmed the same findings.

MOVIE RECOMMENDATION

In real-life decision making, people are often faced with an overload of choices, for instance when purchasing products, choosing a movie to watch or a restaurant to go out to dinner. Within the field of psychology, research has shown that people are less capable of making satisfactory decisions when given more choices and when they are exposed to an overload of information regarding those choices (Dijksterhuis et al., 2006; Messner and Wänke, 2011). The goal of a recommender system is to aid the user by reducing the available choices to a shortlist of items of interest to the user. Real-world examples include the suggestion of movies on Netflix and books on Amazon.

To estimate the best items to recommend to a user, two types of state-of-the-art collaborative filtering methods infer the best recommendations from collected user ratings for the items: nearest neighbor based methods use similarities between users and/or items, and matrix factorization algorithms learn low-dimensional representations (factors) that describe the preference patterns of users towards items. The nearest neighbor based methods are effective but lack scalability. Matrix Factorization typically scales better, however, for effective recommendation they typically require more training data and the factors that are described are abstract and not easily interpreted.

In Part III of this thesis, we study how semantic spaces can be used for item recommendation. When learning semantic representations for words based on the context they appear in, semantic differences between words are encoded consistently, e.g. difference in gender, and countries and their capitals (Mikolov et al., 2013). Similarly, for learned movie representations, we observe that patterns that describe the differences between movies are consistently encoded, e.g. movie genres, suspense. For movies there are many factors that are useful to describe why a user prefers some movies over others (e.g. favorite actor, director, scary elements, drama, humor), and in a low-dimensional space, such as typically used by matrix factorization algorithms, factors are not encoded independently limiting the possibilities to distinguish between them. Therefore we propose to learn high-dimensional item representations.

When recommending items to a specific user, we seek to represent the user's interest over the factors that are encoded in the semantic space. An important consideration is that a user may only care about a limited number of factors. For instance, some users may like or dislike scary elements in movies, while other users are indifferent to whether a movie contains scary elements. Thus, when recommending items, we have two considerations. Firstly, in the semantic space the 'best' recommendation candidates are likely to be positioned close to the items rated highly by the user. Secondly, the vector between two movies in semantic space should be reflected in their ranking by the extent to which the corresponding encoded factors are preferred by the user. To recommend movies to a specific user, hyperplane coefficients are learned to optimally rank the items according to a user's past ratings. An advantage of ranking the items according to their distance to a hyperplane, is that if two movies are separated by factors that the user is indifferent to these can be ignored by choosing a hyperplane that is parallel to the direction of these factors in the semantic space.

For the evaluation, we learn semantic vectors for movies based on the ratings they received from users, and show our approach greatly outperformed existing state-of-the-art recommender algorithms. We show that the proposed architecture can also be used for content-based recommendations, which can potentially be used to improve the recommendation of new or rarely rated items.

I

Term proximity and its use to estimate the relevance of documents

1

Distance matters: Cumulative proximity expansions for ranking documents

In the information retrieval process, functions that rank documents according to their estimated relevance to a query typically regard query terms as being independent. However, it is often the joint presence of query terms that is of interest to the user, which is overlooked when matching independent terms. One feature that can be used to express the relatedness of co-occurring terms is their proximity in text. In past research, models that are trained on the proximity information in a collection have performed better than models that are not estimated on data. We analyzed how co-occurring query terms can be used to estimate the relevance of documents based on their distance in text, which is used to extend a unigram ranking function with a proximity model that accumulates the scores of all occurring term combinations. This proximity model is more practical than existing models, since it does not require any co-occurrence statistics, it obviates the need to tune additional parameters, and has a retrieval speed close to competing models. We show that this approach is more robust than existing models, on both Web and newswire corpora, and on average performs equal or better than existing proximity models across collections.

***Keywords**: Term dependency · Term proximity · Query expansion*

The contents of this chapter has been published as J.B.P. Vuurens and A.P. de Vries, Distance Matters: Cumulative Proximity Expansions for Ranking Documents. Information Retrieval, Springer, 2013.

1.1. INTRODUCTION

Information Retrieval (IR) has changed considerably over the years, due to the expansion of the World Wide Web and an increased use of the Web as the primary source of information. While there are many ways to measure the performance of an IR system, in general, the aim is to rank documents according to relevance perceived by the user. Retrieving relevant documents is a challenging task, because there usually exists no "perfect query" that unambiguously provides a clean match between the user needs and relevant information. In fact, queries typically match more irrelevant documents than relevant ones.

The problem of ranking documents for a given query is usually simplified by treating query terms as being independent, ignoring any user's interest in the joint co-occurrence of query terms such as forming a noun phrase. One feature that can be used to express the relatedness of co-occurring terms is their proximity in text. Intuitively, researchers have suspected that query terms that appear closer together in documents represent stronger evidence for relevance (Clarke et al., 2000; Croft et al., 1991; Keen, 1991; Lv and Zhai, 2009; Metzler and Croft, 2005; Tao and Zhai, 2007; Zhao and Yun, 2009). Early studies used theoretical assumptions about proximity, which provided occasional rather than substantial results (e.g. Croft et al., 1991; Fagan, 1987). More recent proximity models make use of parameters that allow them to fit themselves to the data in the collection and more consistently improve retrieval performance over independent term baselines (Metzler and Croft, 2005; Tao and Zhai, 2007). Although these studies show the potential of term proximity for information retrieval, they provide limited insight into the relation between term proximity and document relevance. To obtain optimal results, these models often use only a selection of term combinations and occurrences within some maximum word distance. As a consequence, there is great diversity between studies upon their assumption how relevance is affected by the distance between query terms (Clarke et al., 2000; Lv and Zhai, 2009; Metzler and Croft, 2005; Tao and Zhai, 2007; Zhao and Yun, 2009).

We hypothesize that the performance of proximity retrieval models can be improved by removing constraints regarding distance and the selection of term combinations. In this study, we specifically reexamine the relationship between the distance of co-occurring query terms and their estimated relevance, to design a Cumulative Proximity Expansions (CPE) retrieval model. We show that this model is more robust and performs equal or better to state-of-the-art proximity models on both Web and newswire collections. This model is also more practical than existing models, since it does not require any co-occurrence statistics, it obviates the need to tune additional parameters, and has a retrieval speed close to competing models.

The paper is structured as follows: Section 2 discusses related work that is relevant to term dependency in information retrieval models. In Section 3, the effects of promoting proximity are examined. Based on this analysis, we describe a proximity model in Section 4. In Section 5, we discuss the implementation, which is available for download, and compare the retrieval speed of the proposed model to five baselines over eleven test sets. Section 6 presents the empirical results of the model over ad-hoc TREC collections and the comparison with state-of-the-art proximity models. The conclusions are presented in Section 7.

1.2. RELATED WORK

This section discusses previous research in the area of term dependencies, specifically: query operators to match term-dependencies in documents, the selection of term combinations used in a dependency model, the relation between the proximity of query terms and the likelihood of being relevant, and how proximity evidence can be used in a function to rank documents.

We first survey, briefly, the literature addressing term proximity for information retrieval models. We then summarize the key aspects with respect to the question of scoring term proximity. The third and final subsection details the baseline approaches chosen for our own empirical work.

1.2.1. TERM DEPENDENCY

Term dependency is a recurring topic throughout the history of information retrieval research, and it is plain impossible to do justice to every single contribution in this area. As we will see, many attempts never materialized in consistent improvements in retrieval performance. However, with today's larger corpora, improvements are feasible and there has been a renewed interest in this topic. We highlight the papers most relevant for our own work in this section.

An early paper to address the effect of term dependency on text retrieval is Van Rijsbergen (1977), describing a theoretical basis for the use of co-occurrence data in ranking documents. For every query term, the most likely dependent other query term is selected based on the expected mutual information measure computed for the pair of terms. Van Rijsbergen proposes to use the resulting Maximum Spanning Tree (MST) for retrieval. Only years later, this idea has been followed up upon when Nallapati and Allan (2002) presented their approach to select the most significant query term dependencies. To reduce the runtime required, the Maximum Spanning Tree was build using sentence statistics rather than document statistics. Results of this study were characterized as "a slight improvement in performance", and we are not aware of later follow-up work.

The rather early empirical study of Fagan (1987) discussed a comparison of methods to select phrases with semantically related terms. Fagan's most successful attempt matched pairs of query terms to document contents, but without consistent improvements in retrieval performance.

Keen (1991) describes results of another early empirical study, using test collection created in 1982 from bibliographic summary records. Keen suggested that using information on term positions can help narrow down search results, by screening out irrelevant results. As far as we know, he was the first to formulate the intuition that motivated our own exploration of term proximity, when he suggested the number of *intervening terms* as the factor determining the strength of the relation between pairs of query terms: "the number of non-matching terms found to lie between the first and last matching terms in the sentence". He explored seven different approaches to make use of the proximity between terms, based on the actual term distance, query term co-occurrence in sentences, and a combination of these two principles (e.g. terms in close proximity within a sentence). All seven methods were demonstrated experimentally to improve retrieval performance over the (by now outdated) baseline, a system using coordination level ranking. In this study, the most effective results were obtained with an algorithm that rewards a low distance between query term pairs.

Croft et al. (1991) first explored how to integrate term co-occurrence information derived from phrases into the inference network retrieval model underlying InQuery (and, much later, Indri). The noun phrases considered were extracted from the information request using a stochastic tagger. Croft et al. further suggested the idea of removing individual query terms with a high collection frequency, and matching these only as constituents of a phrase, e.g. the word "system" in "computer system" and "operating system". They obtained improvements in precision at low recall levels, but results at higher levels of recall were inconclusive. The authors noted that their results suggest a higher contribution to retrieval performance on larger collections, a finding that has been confirmed in later work, using a new retrieval model based on Markov random fields (Metzler and Croft, 2005).

Various more recent studies have introduced term dependencies in the language modeling approach to information retrieval. Song and Croft (1999) first considered extending the "standard" unigram models by interpolation with a model for bi-grams. A small scale experiment indicated improved retrieval effectiveness by using the word pairs. Gao et al. (2004) proposed the Dependence Language Model, a joint distribution between a unigram language model and a dependency model that promotes the documents that contain co-occurring query terms (within a sliding window of three words). From a training corpus they estimated the most likely "linkage" that sequentially connects all query terms using every query term once. Only term-pairs in this linkage were considered in the dependency model. Their model consistently improved retrieval performance on smaller TREC collections; however, according to Metzler and Croft (2005) and He et al. (2011), the requirement to compute the likelihood of all possible link structures for a query may be prohibitive for application in practical retrieval systems.

A different line of research explored how to integrate term dependencies in the classic probabilistic retrieval model. For example, Rasolofo and Savoy (2003) proposed to expand BM25 with a proximity measure, by accumulating a distance score for every co-occurring term pair within a sliding window of five words. Their accumulated distance function replaces the term frequency in a BM25-like function, using the lowest weight of the two terms in the pair. The score of the co-occurring term pairs is then added to the score of the unigrams. Their experimental results show improvements at the top of the ranking (precision at five), for all three test sets used, but with mixed results on other metrics. Song et al. (2008) introduced a different perspective on term proximity, by forming non-overlapping spans of multiple query terms (not just query term pairs). Each span (identified through intuitive heuristic rules) is then assigned a relevance weight, based on the length of the span and the number of query terms contained. These weights are aggregated per query term, replacing the original term frequency of the BM25 ranking formula. He et al. (2011) used a sliding window to count the frequency of n-grams in a document, and modified BM25 to score n-grams containing multiple query terms in a way similar to unigrams. In their survival model, they promoted term dependency using the minimal number of words that separate a sequence that contains all query terms.

Metzler and Croft (2005) introduced the Markov Random Field (MRF) retrieval model, a flexible model that is especially suited for modelling term dependencies. The MRF is constructed from a graph where nodes correspond to query terms, and the edge configuration imposes the independence assumptions made. The authors present three variants, corresponding to the traditional full independence, a new sequential dependence model where neighboring query terms are connected by an edge, and a fully connected variant (where every query node is connected to every other query node). Potential functions defined over the cliques in these graphs determine the final ranking function; that combines linearly a relevance score for independent query terms with a score for ordered term pairs (in the sequential dependence case) and one for unordered term combinations (in the full dependence case). The mixture parameters are tuned by cross-validation (selecting the highest mean average precision (MAP) for each fold). Empirical results show significant improvements by modelling term dependencies explicitly in the MRF. The authors concluded that the Sequential Dependence Model (SDM) is the best choice on smaller but homogeneous collections, with longer queries, while the Full Dependence Model (FDM) attained better results for larger, less homogeneous collections, but using shorter queries.

Metzler and Croft (2007) later expanded the MRF framework with a method to select the most likely "latent concepts" that the user had in mind, but did not express in the query. Similar to the Relevance Model by Lavrenko and Croft (2001) single word concepts are extracted from (pseudo-) relevance feedback documents, to which they add extracted multi-word concepts. The expansion with latent concepts improves the performance significantly over the original MRF model.

Shi and Nie (2010) reflected on Metzler and Croft (2005), arguing that it is not reasonable to expect the same fixed value to score unigrams, term sequences and term co-occurrences within a sliding window. Both Shi and Nie (2010) and Bendersky et al. (2010) therefore extended Metzler and Croft's SDM model with separate parameters for each unigram and each term-pair. This however leads to a huge number of model parameters that require tuning through n-fold cross-validation, for which both approaches build on a coordination-ascent search algorithm, while Shi and Nie (2010) combines this with an approach based on Support Vector Machine regression. Bendersky and Croft (2012) then proposed a model that is reminiscent to MRF, in which three types of linguistic structures are considered: the original query terms, adjacent term-pairs and unordered co-occurrences of selected term combinations within some window size. The ranking function combines independent scoring of all concepts using a log-linear scoring function, extended with the score of a "global hyper edge" that considers each concepts contribution to the entire set of query concepts. A possible advantage of the global hyper edge over the full dependence model by Metzler and Croft (2005) is that it can express a dependency between multi-term concepts, rather than all single terms co-occurring within some distance. A learning-to-rank approach was used to train the large set of parameters. Unfortunately, the improvements in effectiveness were only marginal.

Two fairly recent papers aimed to investigate ranking using term proximity, but far less tightly connected to the derivation of the retrieval model in which the term dependencies are to be integrated. First, Büttcher et al. (2006) proposed what they called a *cumulative model* that calculates separate proximity scores per query term. Their algorithm considered especially its implementation in the inverted file indexing structure that forms the core component of virtually all retrieval systems at the time. While traversing the posting lists considered for a given query, whenever a query term is encountered that is different from the query term last seen, a distance score is added to their respective accumulators. The scores per term are eventually computed as separate evidence, by using the proximity score instead of the term frequency. A similar approach was taken in Tao and Zhai (2007). The authors compared five proximity measures, each returning an aggregated outcome per document, which is then converted into a term weight using a convex decay over the distance function, and added to either the KLD or BM25 retrieval scores. The best performing proximity measure was the minimum distance between any two different query terms occurring in the text, which consistently improved retrieval performance and was shown to give results comparable to those obtained in Metzler and Croft (2005).

Building upon this latest work, Zhao and Yun (2009) presented the Proximity Language Model (PLM). Here, the minimum distance metric of Tao and Zhai (2007) is used between all the query term pairs. The sum of minimal distances is converted into a score that is added directly to each unigram's term frequency in the KLD function. Their results show a higher mean average precision when compared to those of Tao and Zhai, however, on more than 50% of the reported experiments the results did not significantly improve over the KLD baseline.

A few more approaches should still be mentioned, even if this brief discussion can never do justice to the complete history of term dependency in information retrieval.

Lv and Zhai (2009) proposed the Positional Language Model, that builds upon the idea to propagate each occurring term over the word positions in the document (a notion first introduced in de Kretser and Moffat (1999)). The authors use kernel density estimation, to essentially create a separate language model for every word position in a document. Their ranking function therefore initially ranks document positions instead of documents.

Most Learning to Rank (LTOR) approaches use features derived from query term co-occurrences. As an example, Cummins and O'Riordan (2009) used machine learning to develop term-term proximity functions that optimize mean average precision. They considered aggregated distance statistics, such as the average and minimal distance between query terms in the document. In spite of achieving a consistent increase in MAP on the test collections, the improvements were not always found to be statistically significant. In general, the retrieval functions resulting from the machine learning process tend to exceed human comprehension, not really helping us to understand how distance and relevance are actually related; merely confirming the intuition that such a relationship exists.

So far, we have primarily focused on the way term proximity is integrated in retrieval models for document retrieval. Term proximity is of course closely related to topics like passage retrieval, which in general divides documents into passages as the basis for ranking (Liu and Croft, 2002; Tellex et al., 2003). Similar to term proximity, passage retrieval is less likely to promote documents in which the terms appear further apart, but has been criticized by Tao and Zhai (2007) for being more coarse and limited. Other related approaches are XML information retrieval and sentence retrieval, that both capture aspects of term proximity in a similar way to the passage retrieval approaches.

Finally, term proximity can be used in different aspects of a retrieval system, notably by improving the selection of query expansion terms, e.g. Vechtomova and Wang (2006) compared distance functions to improve the selection of query expansion terms. Empirically, the best variant used an inversely proportional function over term distance to rank candidate expansion terms, which outperformed the use of exponential and logarithmic distance functions. In recent work, Miao et al. (2012) also used proximity to improve the selection of expansion terms from feedback documents. They rank candidate terms using word distance with query terms in the feedback documents, using the Hyperspace Analogue to Language (HAL) model. The results consistently improved performance over a baseline of BM25 with Rocchio pseudo-relevance feedback.

1.2.2. DESIGN ASPECTS OF PROXIMITY MODELS

A large variety of approaches of dealing with term proximity in information retrieval has been presented throughout the years. Looking back on all these works, however, the following three research questions should still be considered as unanswered:

1. What is the range within which co-occurrences of query terms should be considered?
2. How should the distance between co-occurring terms be reflected in their respective term weights?
3. How should evidence derived from term proximity be integrated in the retrieval model?

Nearby term co-occurrences have generally been considered to provide stronger evidence of relevance than more distant co-occurrences. The majority of studies has only considered occurrences within short distance from each other. Given that the relation between relevance and the co-occurrence of query terms is not obvious, let alone when these occurrences are far apart, this seems a natural choice. When asking at what range query term co-occurrences may still influence the relevance estimation process, the literature does not provide an answer. Studies like (Croft et al., 1991; Metzler and Croft, 2005; Rasolofo and Savoy, 2003) have only considered terms that co-occur closely, usually within a window of size eight to ten. However, Song et al. (2008) reported an improvement in retrieval effectiveness using a much larger window size of 50. Similarly, Bendersky and Croft (2012) observed that the distance between the terms (that together constitute a higher level concept) may span a much greater distance than the typical sliding window considered in most of the research.

The second open question concerns how to weigh terms that occur in the document using proximity information. A few studies have simply used a constant contribution, irrespective of the distance between the query terms considered (e.g. Croft et al., 1991; Fagan, 1987; Metzler and Croft, 2005). Many researchers have however introduced a method to discount the contribution of co-occurrences based on their distance in text. Table 1.1 summarizes how the distance between query term co-occurrences has been weighted. Here, *span* refers to the number of word positions covered; $\mathcal{K}$ a threshold on span size above which weights are lowered; *windowsize* the maximum span within which co-occurrences are scored; *terms* the number of query terms that make up the co-occurrence; N is the last position in the document; and, i is the word position for which the weight is estimated. The weight accumulates the distance $|i-j|$ between the current

Table 1.1: Various functions for determining the weight of a co-occurrence based on the distance between terms.

$weight = \begin{cases} 1, & \text{if } span < \mathcal{K} \\ \frac{\mathcal{K}}{span}. & \text{otherwise} \end{cases}$	(Clarke et al., 2000)
$weight = \frac{1}{\sqrt{span - 1}}$	(Hawking and Thistlewaite, 1995)
$weight = \frac{1}{(span - 1)^2}$	(Rasolofo and Savoy, 2003)
$weight = windowsize - span + 1$	(Miao et al., 2012)
$weight = \left(\frac{terms}{span}\right)^x \cdot terms^y$	(Song et al., 2008)
$weight = \log\left(\alpha + \mathrm{e}^{-min(span-2)}\right)$	(Tao and Zhai, 2007)
$weight = para^{-min(span-2)}$	(Zhao and Yun, 2009)
$weight_i = \sum_{j=1}^{N} exp\left[\frac{-\left(i-j\right)^2}{2\sigma^2}\right]$	(Lv and Zhai, 2009)

position i and query term occurrences j using a Gaussian kernel. For further details, please refer to the original paper (Lv and Zhai, 2009). The symbols x, y, α, $para$ and σ are free parameters that need to be tuned on the document collection used. The functions considered have in common that they are convex and monotonically decreasing. Most studies have assigned a default score of one to adjacently appearing terms, with the exception of Tao and Zhai (2007), Song et al. (2008) and Lv and Zhai (2009). Which of these functions would be the preferred choice has never been answered satisfactory.

The first two questions may be left unanswered, or only answered implicitly, when relying on a machine learning method to estimate the best choice given training data to tune model parameters. Then, instead of positing a general assumption on how proximity relates to relevance, the scoring function can be adjusted to the collection, without the need to making such choices a priori. Examples include Metzler and Croft (2005); Tao and Zhai (2007); Zhao and Yun (2009); SDM for example distinguishes between contiguous and non-contiguous appearance of terms in a query and document, and estimates the weights to combine these using cross-validation. Tao and Zhai (2007) use a parameter in a convex decay over distance function, allowing the model to adapt the function to more optimal performance.

Instead of following this trend to let the model adapt to training data, this paper does actually attempt to give an explicit statement of how term co-occurrences is expected to influence relevance. Assuming the first two questions can be settled, the proximity evidence still needs to be combined with the other sources of information upon which a document's probability of relevance is estimated, and especially the independent query term occurrences.

The two most common design patterns in the literature surveyed have 1) added proximity evidence directly to the independent query term frequencies, using the original retrieval model *as is* (e.g. Svore et al., 2010; Zhao and Yun, 2009), or 2) scored proximity evidence separately from the other relevance information, mixing the proximity based score with that of the independent term occurrences (e.g. Metzler and Croft, 2005; Tao and Zhai, 2007). Consider the following example. Let query terms "Albert Einstein" occur once and adjacently in two different documents A and B, but with different frequencies for the word "Einstein". If proximity evidence is added to the raw unigram counts, the document with the lowest unigram count will gain the most from adding the co-occurrence information. When viewed as two separate sources of relevance information, both documents would receive the same contribution for the evidence of query term co-occurrences, irrespective of the frequency of the individual query terms.

1.2.3. Proximity baselines

In this study, we will compare the retrieval performance of four proximity baselines that can be considered state-of-the-art based on the results presented. The selected baselines score terms that occur in a document independently using Dirichlet-smoothed language model functions that are rank equivalent to each other. Therefore, comparing the results of each model to the independent term baseline, will reveal how effective each model is in additionally using proximity information.

Zhai and Lafferty (2004) propose to use negative Kullback-Leibler divergence between a query language model and a Dirichlet smoothed language model of a document (KLD), which they reformulated by removing the query entropy which does not affect document ranking. In Equation 1.1, documents D are ranked for a query Q, q_i is a term in Q, $tf_{q_i,D}$ is the frequency of q_i in D, $|D|$ is the number of terms in D, and μ is the Dirichlet smoothing parameter. In Equation 1.2, cf_{q_i} is the frequency of q_i in the collection C and $|C|$ is the number of words in the collection.

$$KLD(Q,D) \equiv \sum_{q_i \in Q} \left[\log\left(1 + \frac{tf_{q_i,D}}{\mu \cdot P(q_i|C)}\right) + \log \frac{\mu}{\mu + |D|} \right] \tag{1.1}$$

$$P(q_i|C) = \frac{cf_{q_i}}{|C|} \tag{1.2}$$

Tao and Zhai (2007) presented a simple baseline for scoring term dependency. Using Equation 1.3, documents are ranked according to the sum of KLD over independent query terms and a proximity function $\pi(Q,D)$. In Equation 1.4, α is a free parameter and $\delta(Q,D)$ is a distance function, for which Tao and Zhai experimented with five variants. In Equation 1.5, $Dis(q_i, q_j; D)$ is a function that returns the minimum distance in word positions between all occurrences of query terms q_i and q_j in D, or, $|D|$ if D does not contain both terms. In other words, δ is defined as the minimum distance between any two occurrences of distinct query terms, the variant which provided the best results in the retrieval experiments of Tao and Zhai.

$$MinDist(Q,D) = KLD(Q,D) + \pi(Q,D) \tag{1.3}$$

$$\pi(Q,D) = \log\left(\alpha + e^{-\delta(Q,D)}\right) \tag{1.4}$$

$$\delta(Q,D) = \min_{q_i,q_j \in Q \cap D, q_i \neq q_j} \left\{Dis(q_i,q_j;D)\right\} \tag{1.5}$$

Zhao and Yun (2009) presented the Proximity Language Model (Equation 1.6), in which they re-estimate the seen word probability $P(q_i|D)$ with respect to the proximity model (Equation 1.7), and assign $\alpha_D \cdot P(q_i|C)$ to unseen terms (Equation 1.6). In their model, the proximity factor adjusts the seen word probability in a document by adding a proximity centrality $Prox_B$ to the term count in the document (Equations 1.7 and 1.8). Besides the symbols already defined for KLD and MinDist, in Equation 1.9, $Dis(q_i,q_j;D)$ returns the minimal distance between all occurrences of term q_i and q_j in D as described by Tao and Zhai (2007), and λ and $para$ are free parameters.

$$\begin{aligned} PLM(Q,D) = & \sum_{q_i \in Q \cap D} P(q_i|Q) \cdot \log \frac{P(q_i|D)}{\alpha_D \cdot P(q_i|C)} \\ & + \log \alpha_D \end{aligned} \tag{1.6}$$

$$P(q_i|D) = \frac{tf_{q_i,D} + \lambda \cdot Prox_B(q_i;D) + \mu \cdot P(q_i|C)}{\mu + |D| + \lambda \cdot \sum_{q_i \in Q} Prox_B(q_i;D)} \tag{1.7}$$

$$\alpha_D = \frac{\mu}{\mu + |D| + \lambda \cdot \sum_{q_i \in Q} Prox_B(q_i;D)} \tag{1.8}$$

$$Prox_B(q_i;D) = \sum_{q_j \in Q, q_i \neq q_j} \text{para}^{-Dis(q_i,q_j;D)} \tag{1.9}$$

Metzler and Croft (2005) proposed to estimate the relevance of documents using the Markov Random Fields framework. A Markov Random Field is constructed from a graph in which nodes (random variables) correspond to the document and the query terms, and edges represent dependencies between these random variables. Due to conditional independence, the joint distribution between these variables can be factored by considering only the cliques in this graph. Therefore, scoring a document boils down to considering separately the sets T, O, and U, of, respectively, query terms treated as independent (T), contiguous query terms (O), and otherwise dependent query terms (U).

Metzler and Croft consider two term dependency variants. In the Sequential Dependence Model (SDM), only adjacent query terms are considered directly dependent, and O and U will consist of all pairs of terms that appear adjacently in the query. In the Full Dependence Model (FDM), all query terms are directly dependent upon each other, O consisting of all contiguous sequences of two or more query terms and U of all combinations of two or more query terms. The resulting ranking function in Equation 1.10 combines the scores for these three sets of cliques using the linear mixture parameters λ_T, λ_O and λ_U.

The independent query terms T are scored using function f_T (Equation 1.11), which is a language modeling estimate smoothed by a Dirichlet prior α_D, where $tf_{q_i,D}$ is the frequency of term q_i in document D, $|D|$ is the number of terms in D, cf_{q_i} is the frequency of the term in the collection, and $|C|$ the total number of terms in the collection. Contiguously appearing query terms O are scored using f_O (Equation 1.12), in which $tf_{\#1(q_i,...,q_j),D}$ denotes the number of times the sequence of terms $q_i,...,q_j$ appears contiguously in D, and $cf_{\#1(q_i,...,q_j)}$ in the collection. Combinations of query terms U are scored using f_U (Equation 1.13), in which $tf_{\#uwN(q_i,...,q_j),D}$ is the number of times all query terms in the clique appear in any order within a window of N words in D, with N set to 4 times the number of terms in the clique, and $cf_{\#uwN(q_i,...,q_j)}$ the frequency of that event in the collection. In Equation 1.14, α_D is the smoothing parameter also described by Zhai and Lafferty (2004). By definition, $\lambda_T = 1 - \lambda_O - \lambda_U$ leaving λ_O, λ_U and μ to be trained as free parameters.

$$\begin{aligned} MRF(Q,D) = \lambda_T & \sum_{(q_i,D)\in T} f_T(q_i,D) + \\ \lambda_O & \sum_{(q_i,...,q_j,D)\in O} f_O(q_i,...,q_j,D) + \\ \lambda_U & \sum_{(q_i,...,q_j,D)\in U} f_U(q_i,...,q_j,D) \end{aligned} \tag{1.10}$$

$$f_T(q_i,D) = \log\left((1-\alpha_D)\frac{tf_{q_i,D}}{|D|} + \alpha_D\frac{cf_{q_i}}{|C|}\right) \tag{1.11}$$

$$f_O(q_i,...,q_j,D) = \log\left((1-\alpha_D)\frac{tf_{\#1(q_i,...,q_j),D}}{|D|} + \alpha_D\frac{cf_{\#1(q_i,...,q_j)}}{|C|}\right) \tag{1.12}$$

$$f_U(q_i,...,q_j,D) = \log\left((1-\alpha_D)\frac{tf_{\#uwN(q_i,...,q_j),D}}{|D|} + \alpha_D\frac{cf_{\#uwN(q_i,...,q_j)}}{|C|}\right) \tag{1.13}$$

$$\alpha_D = \frac{\mu}{\mu + |D|} \tag{1.14}$$

A final remark on the relationship between the scoring of independent terms in the MRF and KLD baselines. Although the function used to score the independent query term occurrences has a different form in Equation 1.10 than the KLD function described in Equation 1.1, both functions are Dirichlet smoothed language model estimates and are in fact rank equivalent if used with the same value for μ (refer to Appendix A for a derivation of this equivalence).

1.3. ANALYSIS

To examine term proximity, we used an oracle system on the TREC-6 ad-hoc topics to improve the retrieval performance by expanding the queries with proximity operators. We describe how the oracle system maximizes results, report the reformulation for the most improved queries and discuss the results. We then continue to analyze how the relevance of documents can be estimated based on the distance between co-occurring query terms.

1.3.1. ORACLE EXPERIMENT

Most existing proximity models make limited use of the available proximity information in documents, considering only a selection of term combination and occurrences

Query: best basketball player	
Document 1	: ... best basketball player ...
Document 2	: ... best training exercise for basketball players ...
Document 3	: ... best basketball for beginning players ...

Figure 1.1: The query is more likely to be satisfied by Document 1 than by Document 2 or 3

within some maximum word distance. As far as we know, previous studies do not explain why more distance between co-occurring query terms represent weaker evidence for a document being relevant, or whether it is justified to assume that proximity is only useful within a limited word distance. We hypothesize that words in between query term occurrences weaken their relatedness. Therefore, proximate terms are more likely to appear in relevant documents than distant ones. In Figure 1.1, we illustrate this intuition by sentences that would each match the hypothetical query "best basketball player". Although in relevant documents these terms do not necessarily appear consecutively, more distance between the query terms provides more opportunity to divert or weaken the relation between them, e.g. "best training exercise for basketball players" or "best basketball for beginning players". Since we expect an increase in the number of intermediate words to increase the likelihood of weakening the relation between query terms, the relation between proximity and relevance may neither be limited to some distance, nor only apply to some term combinations.

To analyze the potential of using proximity in ranking documents, we constructed an oracle system which performed a simple breadth-first search to optimize queries by adding proximity operators using two or more query terms, evaluating each variant using known relevance judgments. The system used all possible query-term combinations as potential proximity expansions, and was allowed to independently adjust each proximity operator by setting a $weight \in \{1.0, 0.50, 0.25\}$ and a maximum $span \in \{2, 3, 4, 5, 10, 20, 50, 100, 200, 500\}$. The original query was used as the initial best query. For this query, all previously unseen single variations were tried, i.e. adding, removing or modifying only one proximity expansion. The best query was then replaced by the variant with the highest mean average precision (or set of variants when tied) and used as input for the next iteration to try new variants. This was repeated until there were no more untried single variations to the best query, thus converging to a (local) optimum.

We used the oracle system to improve the TREC-6 queries with proximity expansions, and list the 10 queries that were improved most in Table 1.2. Documents were scored according to Equation 1.15, which uses KLD to score the independent terms, and KLD_{co} to score the proximity expansions M. In Equation 1.16, m_i is the i-th proximity expansion in M, λ_i is the weight for m_i, $tf_{\#uw\delta_i(m_i),D}$ is number of occurrences matched by m_i in document D, matching the terms in any order within a window of δ_i word positions. In Equation 1.17, $cf_{\#uw\delta_i(m)}$ counts the occurrences of m_i in the entire collection and $|C|$ is the number of word in the collection. To explain syntax used in Table 1.2, "{air pollution span=200}#0.5" scores all unordered co-occurrences of "air" and "pollution, using $\delta = 200$ and $\lambda = 0.5$. Over the TREC-6 test set, the oracle system improved over the KLD baseline by 17.8% and over the SDM baseline by 13.3%, indicating the potential contribution for term proximity to estimate document relevance exceeds that of the SDM

Table 1.2: 10 TREC-6 queries that gained most (in AP%) in the Oracle experiment.

topic	original query	oracle expansion	KLD AP	SDM AP	Oracle AP

303	Hubble Telescope Achievements	{hubble telescope span=2}
		{hubble achievements span=5}
			0.205	0.231	0.342

308	implant Dentistry	{implant dentistry span=100}#0.25	0.480	0.479	0.554
310	Radio Waves and Brain Cancer	{radio waves cancer span=100}#0.25 {brain cancer span=2}#0.5	0.160	0.185	0.242
311	Industrial Espionage	{industrial espionage span=10}#0.5	0.372	0.576	0.635
320	Undersea Fiber Optic Cable	{undersea fiber span=2}#0.5 {undersea optic span=10} {fiber cable span=2}#0.5 {undersea fiber cable span=3}#0.25 {optic cable span=20} {undersea optic cable span=5} {fiber optic cable span=3}#0.25	0.023	0.028	0.138

329	Mexican Air Pollution	{mexican air span=500}
		{mexican pollution span=500}
		{air pollution span=200}#0.5
		{mexican air pollution span=20}
			0.141	0.125	0.304

331	World Bank Criticism	{world bank span=2} {world criticism span=200}#0.25 {world bank criticism span=20}#0.5	0.213	0.287	0.419

332	Income Tax Evasion	{income tax span=2}#0.25
		{income evasion span=10}
		{tax evasion span=2}
		{income tax evasion span=3}
			0.126	0.139	0.342

341	Airport Security	{ airport security span=50}	0.232	0.282	0.329
350	Health and Computer Terminals	{health computer span=200}#0.5 {computer terminals span=2} {health computer terminals span=500}	0.105	0.116	0.169

baseline. We make three observations: (1) for queries with three or more terms, the best solution often uses several overlapping proximity operators, e.g. “undersea fiber cable” along with “fiber cable”; (2) using a maximum allowed span of more than 100 words is most effective for some term combinations; (3) the best expansion often uses lower weights for the co-occurrences than for the original query terms.

$$Score(Q, M, D) = KLD(Q, D) + KLD_{co}(M, D) \tag{1.15}$$

$$KLD_{co}(M, D) = \sum_{i=1}^{|M|} \lambda_i \cdot \log\left(1 + \frac{tf_{\#uw\delta_i(m_i),D}}{\mu \cdot P(m_i|C)}\right) \tag{1.16}$$

$$P(m_i|C) = \frac{cf_{\#uw\delta_i(m_i)}}{|C|} \tag{1.17}$$

1.3.2. THE RELATION BETWEEN DISTANCE AND RELEVANCE

In Section 1.2.2, we reviewed existing proximity functions and noticed that in general a value of one is assigned to adjacently appearing terms, occurrences that appear non-contiguously in text receive a lower value based on a linear-convex function over their span in text, and co-occurrences whose span exceeds a maximum distance are ignored. To design a proximity function that benefits from using co-occurrences over longer distance, we argue that the score of both distant and nearby occurrences should reflect their probability to occur in a relevant document. Beeferman et al. (1997) have shown that the co-occurrence frequency between words decays exponentially over their distance. In this study, we are not interested in how often terms co-occur, but rather in the likelihood that the document they occur in is relevant. We analyzed this in a similar experiment, by using all possible combinations of two or more terms from the TREC 1-3 & 5-8 ad-hoc queries, which were counted separately for relevant and irrelevant documents in the corresponding corpora using the TREC qrels. We estimate the likelihood that an occurrence appears in a relevant document given the number of separating terms using Equation 1.18, for which we define $d = R$ as a predicate to test that document D is relevant for the given query, m is a combination of two or more terms from query Q as defined by the Power Set $\mathscr{P}_{>1}(Q)$, and the function $tf_{\#ew\delta(m,d)}$ counts all non-overlapping unordered occurrences of m that are separated by exactly δ terms in D. The obtained frequencies were smoothed using a Gaussian kernel with a bandwidth of $1 + (\delta/2)$.

$$P(d = R|\delta) = \frac{\sum_Q \sum_{d\in C:d=R} \sum_{m\in\mathscr{P}_{>1}(Q)} tf_{\#ew\delta(m,d)}}{\sum_Q \sum_{d\in C} \sum_{c\in\mathscr{P}_{>1}(Q)} tf_{\#ew\delta(m,d)}} \tag{1.18}$$

In Figure 1.2, the “all combinations” line is the result of this experiment, which unexpectedly increases monotonically after reaching a minimum value. Inspection of the data revealed that some term combinations exhibit the opposite behavior, such that close proximity is inversely proportional to the probability of relevance. Ideally, we would like to be able to a-priori identify term combinations that have this inverse distance/relevance relationship, but this is not an issue we will resolve in this study. However, we expect that the undesirable effect of these inverse combinations is partly suppressed by the presence

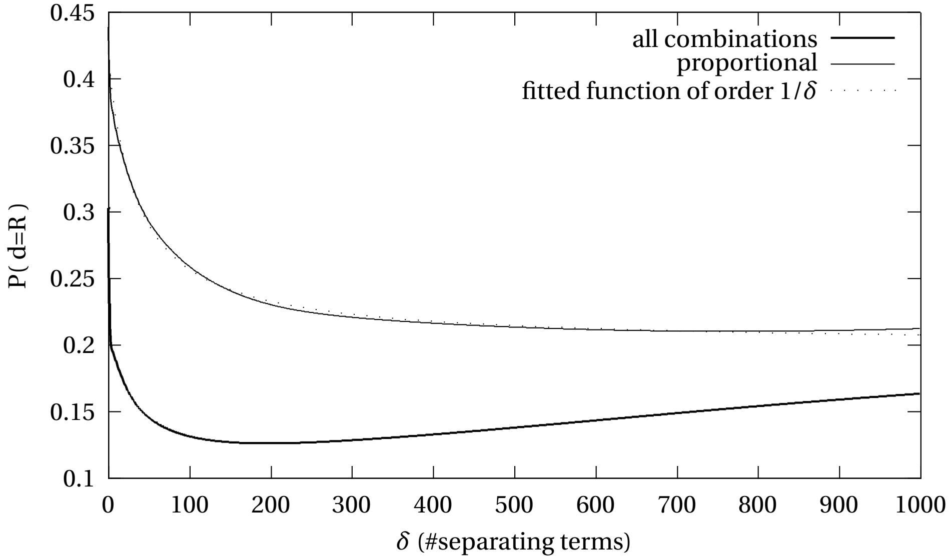


Figure 1.2: The probability of term combinations to appear in a relevant document given the number of separating words, estimated from the TREC 1-3,5-8 ad-hoc topics. The results are compared between "all combinations" and "proportional" term combinations that are more likely to appear in a relevant document at close proximity. Fitted to the proportional set is a function of the order $1/\delta$.

of other query terms, which provide a context within which the inverse combinations may behave differently. In order to focus on term combinations that are more relevant at closer distance, we simply identified inversely related term combinations as having a higher average probability of relevance at 500-1000 terms distance than for 200-500 terms distance. For the set that is labeled "proportional" in Figure 1.2, these inversely related term combinations were removed. For the "proportional" set, the probability of appearing in a relevant document over the distance between terms can be approximated with an inversely proportional function, which is drawn as a dotted line in Figure 1.2 with a goodness-of-fit of $R^2 > 0.99$. From these results, we make two observations: (1) adjacently appearing query terms are roughly twice as likely to appear in relevant documents than query terms that are far apart, and (2) the probability of appearing together in a relevant document can be estimated using the distance between terms using a function of the order *1/distance*.

1.4. METHODS

In Section 1.2.2, we reviewed existing proximity functions and concluded that several aspects have remained unresolved. In Section 1.3.1, we analyzed the potential of proximity operators to improve retrieval performance and estimated the likelihood that a document is relevant given the word distance between occurring term co-occurrences. We now continue to design a proximity retrieval model that uses these aspects. A special variant of this proximity retrieval model also considers the stop words, which are often ignored by retrieval models.

1.4.1. CUMULATIVE PROXIMITY EXPANSIONS MODEL

The Cumulative Proximity Expansions model (CPE) we propose is a simple and, as we will show, effective retrieval model. CPE expands the KLD function for independent query terms with a proximity model that scores every possible combination of two or more query terms. To identify occurrences of term combinations in a document, a proximity operator is used that returns the maximum number of the shortest possible non-overlapping text passages that contain all specified terms in any order. For overlapping occurrence candidates, the shortest or left most candidate is selected first. Different term combinations are scored independently of each other, therefore occurrences of different term combinations can have overlapping word positions, e.g. occurrences of the combination “dog law enforcement” can overlap with occurrences of the combination “law enforcement”, which results in a higher estimated relevance of a document in which subsets are co-occurring more closely.

Based on the dependency we found between the distance of co-occurring query terms and the likelihood to appear in a relevant document in Section 1.3.2, we formulate the following requirements to a function to score term co-occurrences: (1) double the score contribution of query terms occurrences if they appear contiguously in a document, similar to the observed likelihood in Figure 1.2, (2) let the score increment of non-contiguous query term combinations in documents decay with a function of the order 1/distance, and (3) assign the same score for occurrences that are relatively separated to the same extent. The third requirement helps to normalize the scores between term-combinations of different sizes; to clarify, the relative separation of a 2-term combination separated by 2 other words should be weighed similar to a 3-term combination separated by 4 other words in text, having the same ‘density’. In Equation 1.19, we score a term combination m for document D, by using a function similar to the KLD function over the contained terms. We estimate the frequency $tf_{m,D}$ in Equation 1.20 using a proximity operator $\#uw(m,D)$ to match all occurrences o of m in D, and aggregate the frequency per term combination using a $1/distance$ function based on the number of terms $|m|$ and the span of occurrence o in the document. This score function therefore satisfies all three requirements.

$$PROX(m,D) = \sum_{q_i \in m} \log\left(1 + \frac{tf_{m,D}}{\mu \cdot P(q_i|C)}\right) \tag{1.19}$$

$$tf_{m,D} = \sum_{o \in \#uw(m,D)} \frac{|m|-1}{|o|-1} \tag{1.20}$$

Documents are then ranked according to Equation 1.21, which combines the KLD baseline with all combinations of two or more query terms m as defined by the power set $\mathscr{P}_{>1}(Q)$ over query Q from which stop words are removed. To balance the score of the term combinations with respect to the score of independent terms, we introduce a weight function Z, which is necessary to compensate because an additional query term will linearly increase the number of scored independent terms, while the number of scored term combinations grows exponentially. However, we expect the scoring mass of the term combinations to grow as a function over $|Q|$ rather than to grow exponentially, because combinations are less likely to occur as they contain more terms. Empirically, we found that Equation 1.22 gives stable and close to optimal results.

$$CPE(Q,D) = KLD(Q,D) + \frac{1}{Z} \sum_{m \in \mathscr{P}_{>1}(Q)} PROX(m,D) \tag{1.21}$$

$$Z = |Q| \tag{1.22}$$

1.4.2. STOP WORDS IN THE PROXIMITY MODEL

The default proximity model (Equation 1.21) uses the power set $\mathscr{P}_{>1}(Q)$, where Q is the query from which stop words are removed. However, in proximity models stop words could be more useful than in independent term models, promoting multi-word concepts (e.g. “The Beatles”) or passages containing an intended relation nearby targeted terms (e.g. “boy likes girl”, “parents against education”). We hypothesize that considering the stop words in the proximity model may improve retrieval performance. We will test this using a variant of CPE called CPES, in which we replace the power set in Equation 1.21 with a power set over a query with stop words. The set of term combinations is cleaned by removing combinations with stop words that are less likely to be relevant. For this, we introduce a simple heuristic: a combination of query terms is ‘valid’ when the stop words considered are used in combination with all terms that connect them in the query to a non stop word on the left and right, or the query boundary if there is no more non-stop-word. For example, “The Beatles on a zebra crossing” would generate besides “Beatles zebra”, “zebra crossing”, “Beatles crossing” and “Beatles zebra crossing”, the following combinations containing stop words: “The Beatles”, ‘The Beatles zebra”, “The Beatles crossing”, “The Beatles zebra crossing”, “Beatles on a zebra”, “Beatles on a zebra crossing”, “The Beatles on a zebra” and “The Beatles on a zebra crossing”.

1.5. IMPLEMENTATION

To improve reproducibility, we carried out all experimental evaluation using a general purpose retrieval framework, and make all code publicly available, as described in this Section. We then continue to discuss optimizations in the implementation of CPE and CPES, to achieve acceptable speed while scoring all term combinations.

1.5.1. OPEN SOURCE

To facilitate reuse of retrieval components and to improve reproducibility, we have created a general purpose framework called RepIR (Repository for Information Retrieval experiments), which uses Hadoop to extract features from a collection and store these in a central repository, which can then be easily accessed for analysis and retrieval tasks. To compare CPE against existing models using the same index, all models listed in Table 3 have been implemented in RepIR according to their specification. To reproduce our results, RepIR can be downloaded as open source or as a Maven project from http://repir.github.io/. This repository also hosts the specific implementations for this paper along with configuration files and tuned parameter settings.

Table 1.3: Retrieval models implemented in RepIR for this study

Model	Java classes in package io.github.repir.Strategy
KLD	RetrievalModel, ScoreFunctionKLD
CPE	CPERetrievalModel, CPEFeature, ScoreFunctionKLD
CPES	CPESRetrievalModel, CPESFeature, ScoreFunctionKLD
MinDist	MinDistRetrievalModel, MinDistFeature, MinDistScoreFunction
PLM	PLMRetrievalModel, PLMFeature, PLMScoreFunction
SDM	SDMRetrievalModel, SDMTerm, SDMOrderedPhrases, SDMUnorderedPhrases, ScoreFunctionDirichletLM
FDM	FDMRetrievalModel, FDMTerm, FDMOrderedPhrases, FDMUnorderedPhrases, ScoreFunctionDirichletLM

1.5.2. RETRIEVAL SPEED

The CPE model uses all possible query term combinations to compute a relevance score for documents given a query. Expanding the query with an independent proximity operator for every term combination, will generate an exponential number of elements to score and thus is not feasible for long queries. However, in general, terms co-occur far less frequently than that they occur as independent terms, becoming more rare when more terms are combined. Given these distributional characteristics, retrieving can be inefficient when each document is independently inspected and scored for every term combination, even for combinations that are not present. Since the KLD function assigns a score of zero to terms that are not present, this simplifies handling term combination that are not present by simply not scoring these. By simultaneously traversing all term-position lists for a document in one pass, counting and scoring only co-occurrences that are encountered in the document, the number of unnecessary operations is reduced, especially for long queries. Instead of using separate proximity operators, we therefore implemented CPE as one module, which combines the scoring of all term combinations, to allow document processing as described.

CPES will be considerably less efficient than CPE, not only because it additionally uses stop words but also because stop words have long postings lists. However, we prune the processing of stop words using the heuristic we presented in Section 1.4.2, by monitoring if the document contains all terms needed to score a combination with a stop word and otherwise stop traversing the positions list of that stop word for that document. For example, the stop word “on” in term combination “the Beatles on a Zebra Crossing” needs only be used for documents that also contain the words “Beatles”, “a” and “Zebra”. Therefore, if we pass the last occurrence of the word “Beatles” in a document, we can stop traversing the stop words “the”, “a” and “on”. As a consequence, a document is not even scored if it only contains the stop words in the query but no other terms, for not meeting the criteria for any scorable unit.

We analyzed the feasibility of using all term combinations on long queries, using the TREC-4 ad-hoc descriptions (except topic 225, for which time measurement was heavily affected by excessive garbage collection). The descriptions in TREC-4 on average contain 17 terms (stop words included) and probably contain more stop words and non-discriminative words than real user queries. We must note that our implementation is not well suited for accurate comparison of retrieval speed; not being optimized for interactive retrieval, operating on a public Hadoop cluster which makes time measurements inaccurate, and using a single index with positional postings lists for all models, which is slower than necessary for KLD which does not need term positions. To obtain a reasonable result given these circumstances, retrieval is executed within a single mapper, measuring the time within the mapper between the start and end of retrieval to eliminate as much time lost to Hadoop overhead as possible, and by using the fastest retrieval time per query per model of 100 repeated executions, which is the time that is least likely to be affected by external delays. The collection statistics for term co-occurrences that are required for SDM were pre-collected and read into memory, therefore taking no extra time in this test.

In Figure 1.3, we compare the relative retrieval speeds of SDM, CPE and CPES, measured in times slower than KLD. We do not compare against FDM, for which our implementation that uses simple expansion of the query with proximity operators is neither optimal nor feasible on these long queries. We sorted the topics on the relative retrieval speed of CPES. On average, SDM is 1.7 times slower than KLD, CPE is 1.9 times slower and CPES is 4.7 times slower. For example, one of the longest topics in TREC-4 is 211: “How effective are the driving while intoxicated (DWI) regulations? Has the number of deaths caused by DWI been significantly lowered? Why aren’t penalties as harsh for DWI drivers as for the sober driver.”, which contains 16 non stop words and 17 stop words. The fastest retrieval times measured for this topic using KLD was 1.2s, SDM 3.0s, CPE 3.0s and CPES 7.1s.

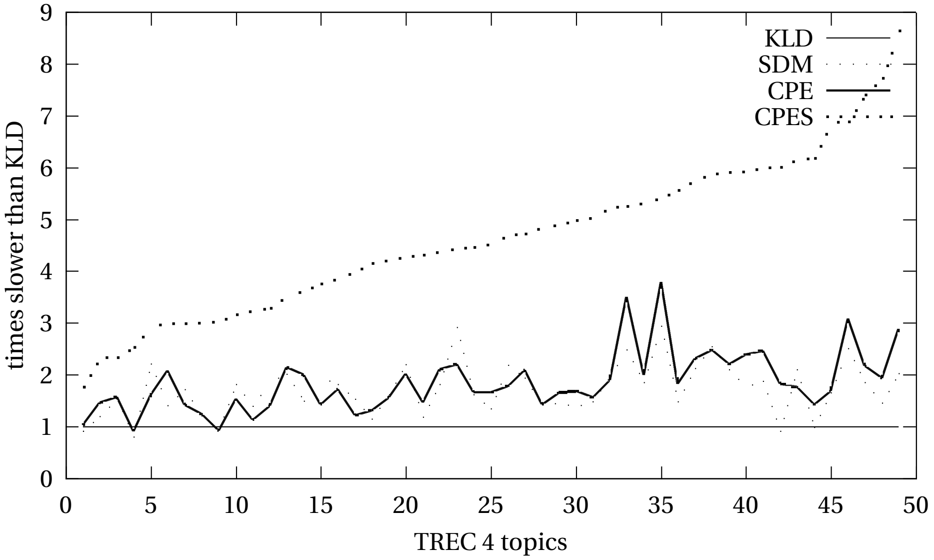


Figure 1.3: The relative retrieval speed of SDM, CPE and CPES, measured in times slower than KLD over the TREC-4 ad-hoc descriptions

1.6. RESULTS AND DISCUSSION

In this Section, we first describe the test sets, collections and parameter tuning, then compare the empirical results obtained with all models, discuss the robustness of the retrieval models, analyze the effects of limiting the span of used co-occurrences, and lastly present a qualitative analysis.

1.6.1. EVALUATION SETUP

For this study, the default repository builder in RepIR was used to create separate positional unigram indexes of the document collections as described by TREC for the TREC 1-3, 5, 6, 7-8 ad-hoc tasks, the English section of ClueWeb09 and ClueWeb12 for the Web Track ad-hoc tasks. During extraction, Unicode and HTML special codes were converted to their ASCII equivalents, and all other Unicode and HTML code was removed with the exception of the contents of “alt” attributes and the meta tags “keywords” and “description”. The remaining content was lowercased, tokenized and stemmed with an English Porter-2 stemmer. The vocabulary size was reduced for ClueWeb09 and ClueWeb12 by leaving out numbers as well as words with more than 25 characters, and by leaving out infrequent words; i.e. that appear less than 2 times on TREC 1-8, and less than 10 times on ClueWeb09 and ClueWeb12. Stop words were indexed thus affecting the positions of non stop word in documents, however, words that are in the SMART list of stop words were not scored during retrieval, except for the experiment that used stop words in the proximity model.

For retrieval, only the titles of the topics were used from the TREC 1, 2, 3, 5, 6, 7, 8 ad-hoc tasks (TREC sets), and the TREC Web Track 2009, 2011, 2012 and 2013 ad-hoc tasks (Web Track sets). Two sets were not used: TREC 4 which contains unrealistic long descriptions as topics for ad-hoc tasks, and Web Track 10, in which 28 out of 50 queries were not useful for evaluation[1]. For the evaluation, all queries in the test sets were used, including the queries that only contain one term, with the exception of queries with no relevant documents in the TREC qrels. Retrieval performance was measured in mean average precision (MAP) for the TREC sets and Statistical Mean Average Precision (StatMAP) for the Web Track sets (Carterette et al., 2008), using the top-1000 retrieved documents.

The free parameters of the proximity models we compare against were tuned using grid search with the following cross-validation strategies: leave-a-test-set-out when several test sets share the same collection (TREC 1-3, 7-8, Web Track 9,11,12), and 10-fold cross validation if there is only one test set for a collection (TREC 5, 6, and Web Track 13). The following parameter ranges were scanned to find close to optimal settings: $\mu = 100, 200, ...5000$, MinDist $\alpha = 0.1, 0.2, ..., 1.5$, PLM $\lambda = 1, 2, ..., 10$ and $para = 1.1, 1.2, ..., 2.5$, and SDM $\lambda_O = 0.04, 0.06, ..., 0.20$ and $\lambda_U = 0.02, 0.03, ..., 0.80$. Although Metzler and Croft (2005) suggested that the fuzzy matches of unordered windows deal better with the noise inherent in Web documents, we did not expect that λ_U would tune to higher settings than λ_T, but on ClueWeb09 $\lambda_U = 0.79$ provided the highest and most stable results. Since the presence of stop words in our index increases the distance between other words, this

[1] These queries contained only one non stop word or did not have any relevant documents in the relevance judgments.

may have positively affected the importance of matched unordered term combinations. However, it is unclear if this indeed the cause for this anomaly.

For KLD, we compared the performance when using $\mu = 2000$ as advocated by Zhai and Lafferty (2004) to tuning μ using cross-validation. On average, the advocated setting performed better, specifically Web Track 2012 was an extreme outlier obtaining +50% StatMAP for the advocated settings over training using Web Track 2009 and Web Track 2011 combined. Therefore, to obtain a more realistic impression of the benefits of using proximity models on Web collections, we decided to use $\mu = 2000$ for KLD CPE and CPES.

For the ClueWeb collections, the runtime for tuning was significantly reduced by using a smaller subset of the collections, consisting of all documents assessed by TREC and all documents retrieved in the top-10k for all topics and all retrieval models using default parameters. The indexes created for these subsets used the unigram and co-occurrence statistics of the entire collection, so that retrieving the top-1k documents on the subset is near identical to retrieving the top-1k documents on the whole ClueWeb collection. The ClueWeb09 subset contained 1% of the documents in the entire collection and the ClueWeb12 collection 0.1%. Checking the results of the parameter tuning, more than 99.9% of the documents retrieved using the tuned parameters for the Web Tracks sets were contained in the collection used for tuning. Using the tuned parameters, the average difference in StatMAP between retrieving a query's results from the subset using the tuned parameters and from the entire collection corresponded to 0.003%, caused by a very small number of documents that were retrieved from the full collection after training, but missing from the subset.

1.6.2. Results

Table 1.4 presents a side-by-side comparison of CPE and CPES to KLD, MinDist, PLM, SDM and FDM over the TREC sets. On average, FDM and CPE perform better than the other models. The CPES model scores specifically worse than the other models on TREC 1 and 3. We suspect that the effectiveness of using stop words depends on the function these stop words have in the query. In some queries, the stop words are vital clues to recognizing the intended meaning, and thus as a word that should appear in relevant documents. In early TREC editions, topics appear to be phrased as natural language, in which stop words often play a more syntactical role than that they are good predictors of relevant documents. For example, in TREC 3 topic 198:“Gene therapy and its benefits to humankind“, the words “and” and “its” may not be here because the user predicts that documents that are relevant contain these words in greater quantity than documents that are not. Using stop words is more likely to help for WT09 topic 42:“the music man”, more strongly promoting the intended meaning, which is a musical and not “Music Man” which is a manufacturer of musical instruments.

Table 1.5 shows a comparison of all models over the Web Track sets. The results show that SDM, FDM, CPE and CPES are relatively more effective than KLD on the Web Track sets than on the TREC sets, which was also suggested by Metzler and Croft (2005). We suspect that the independent term model is negatively affected by the noise and spam that is present in Web collections resulting in a lower baseline, and that the proximity model is less affected by noise and spam by using the distance between co-occurring terms. The results show that FDM performs best on WT09 and WT11, and that CPES

Table 1.4: Comparison of CPE and CPES to KLD, MinDist, PLM, SDM and FDM over ad-hoc TREC sets. The percentages represent improvement in MAP over the KLD baseline. These improvements were tested for significance with a paired Student's T-test, one tailed, $\alpha = 0.05$. The highest MAP per test set is in bold.

		KLD	MinDist	PLM	SDM	FDM	CPE	CPES
TREC 1	MAP	0.2247	0.2396	0.2432	**0.2436**	0.2433	0.2425	0.2373
	delta		+6.6%	+8.2%	+8.4%	+8.3%	+8.0%	+5.6%
	p-value		0.014	0.011	0.026	0.035	0.030	0.103
TREC 2	MAP	0.2065	**0.2145**	0.2128	0.2125	0.2109	0.2142	0.2141
	delta		+3.9%	+3.1%	+2.9%	+2.1%	+3.7%	+3.7%
	p-value		0.032	0.071	0.104	0.194	0.026	0.035
TREC 3	MAP	0.2758	0.2872	0.2797	0.2924	**0.2925**	0.2840	0.2786
	delta		+4.1%	+1.4%	+6.0%	+6.0%	+3.0%	+1.0%
	p-value		0.007	0.266	0.006	0.004	0.023	0.286
TREC 5	MAP	0.1526	0.1626	0.1790	0.1775	**0.1869**	0.1815	0.1801
	delta		+6.5%	+17.3%	+16.3%	+22.5%	+19.0%	+18.0%
	p-value		0.107	0.045	0.076	0.045	0.032	0.039
TREC 6	MAP	0.2280	0.2396	**0.2437**	0.2384	0.2400	0.2411	0.2418
	delta		+5.1%	+6.9%	+4.6%	+5.3%	+5.7%	+6.0%
	p-value		0.035	0.007	0.047	0.034	0.020	0.017
TREC 7	MAP	0.1937	0.2073	0.2073	0.2060	0.2071	0.2087	**0.2122**
	delta		+7.0%	+7.0%	+6.4%	+6.9%	+7.8%	+9.5%
	p-value		0.024	0.019	0.053	0.042	0.014	0.005
TREC 8	MAP	0.2522	0.2591	0.2567	0.2567	0.2572	0.2621	**0.2633**
	delta		+2.8%	+1.8%	+1.8%	+2.0%	+3.9%	+4.4%
	p-value		0.078	0.139	0.205	0.184	0.023	0.014
Average	delta		+5.0%	+5.8%	+6.1%	+6.8%	+6.6%	+6.1%

Table 1.5: Comparison of CPE and CPES to KLD, MinDist, PLM and SDM over ad-hoc Web Track collections. The percentage represent improvement in MAP over the KLD baseline. These improvements were tested for significance with a paired Student's T-test, one tailed, $\alpha = 0.05$. The highest MAP per test set is in bold.

		KLD	MinDist	PLM	SDM	FDM	CPE	CPES
WT09	MAP	0.0334	0.0358	0.0348	0.0439	**0.0460**	0.0416	0.0425
	delta		+7.1%	+4.1%	+31.2%	+37.4%	+24.5%	+27.1%
	p-value		0.192	0.294	0.017	0.014	0.026	0.017
WT11	MAP	0.0775	0.0920	0.0884	0.1306	**0.1416**	0.1244	0.1381
	delta		+18.8%	+14.1%	+68.7%	+82.8%	+60.7%	+78.3%
	p-value		0.000	0.011	0.002	0.000	0.000	0.000
WT12	MAP	0.0418	0.0277	0.0304	0.0437	0.0481	0.0495	**0.0545**
	delta		-33.7%	-27.4%	+4.5%	+14.9%	+18.2%	+30.4%
	p-value		0.042	0.048	0.342	0.077	0.001	0.000
WT13	MAP	0.1332	0.1387	0.1344	0.1483	0.1441	0.1556	**0.1748**
	delta		+4.1%	+0.9%	+11.3%	+8.2%	+16.8%	+31.2%
	p-value		0.002	0.146	0.003	0.143	0.001	0.000
Average	delta		+2.9%	+0.7%	+28.2%	+32.8%	+29.8%	+43.4%

outperforms the other models on WT12 and WT13, mostly significant. The behavior of PLM is less predictive on the Web Track sets than on the TREC sets. In Section 1.2.2, we mentioned that PLM adds proximity information to the frequency of seen terms, rather than scoring this as separate evidence, which may explain why this model does not transfer well to Web collections, where document length varies, affecting how proximity is scored. WT09 has one query in particular on which all proximity models hurt performance: topic 46 "Alexian Brothers Hospital", for which the KLD baseline retrieves three relevant near-identical documents ranked 2-4, titled "Alexian Brothers Health Center: Contact Information". All proximity models rank these three documents lower for not having "Hospital" close to the other terms. Finally, notice that tuning parameters for SDM on WT13 using 10-fold cross validation resulted in settings for λ_U that vary from 0.13 to 0.46 between the folds, which could indicate that cross validation may not have resulted in close to optimal parameter settings being used.

The results that were presented in Section 1.6.2, were also tested on significant improvements between systems. In Table 1.6, each cell contains the test sets on which the models in the row label performed significantly better than the model in the column label. As a reference, Appendix B contains the highest MAP score obtained by a system that participated in the TREC ad-hoc tasks, and highest StatMAP obtained by a system that participated in the Web Track ad-hoc tasks.

1.6.3. ROBUSTNESS

In this section, the robustness of the proximity models is compared. Here, robustness of a model is defined as the number of queries that is improved rather than hurt. Sakai et al. (2005) introduced the Reliability of Improvement metric (RI), which was later called Robustness Index by Collins-Thompson and Callan (2007). In Equation 1.23, the robustness index RI is a value between -1 and 1, computed as the difference between the num-

Table 1.6: Comparison of significant improvements between six proximity baselines. The collection numbers 1-8 for TREC 1-8 and 9-13 for Web Track 2009-2013 indicate that the proximity model in the row label significantly performed better than the proximity model in the column label. Improvements were tested for significance with a paired Student's T-test, one tailed, $\alpha = 0.05$.

Row > Column	MinDist	PLM	SDM	FDM	CPE	CPES
MinDist		9 11 13				3
PLM	5 6		6			
SDM	9 11 12 13	3 9 11 12 13				3
FDM	5 9 11 12	3 9 11 12	5 11		3 9 11	3
CPE	5 9 11 12 13	8 9 11 12 13	8	8 13		3
CPES	5 8 9 11 12 13	8 9 11 12 13	8 12 13	8 13	11 12 13	

Table 1.7: Robustness Index (RI) of CPE and CPES to KLD, MinDist, PLM and SDM compared to the KLD baseline over ad-hoc collections. In bold is the highest RI per test set.

	MinDist	PLM	SDM	FDM	CPE	CPES
TREC 1	0.18	**0.22**	0.14	0.08	0.18	0.12
TREC 2	0.16	0.10	0.08	0.16	**0.20**	0.16
TREC 3	0.12	0.28	**0.32**	0.28	0.22	0.10
TREC 5	0.24	**0.36**	0.16	0.24	0.34	0.30
TREC 6	0.06	**0.26**	0.14	0.10	0.16	0.12
TREC 7	**0.28**	0.06	0.06	0.08	0.18	0.18
TREC 8	0.00	-0.04	0.12	0.20	**0.24**	**0.24**
WT09	0.04	0.00	**0.33**	0.24	0.18	0.24
WT11	0.40	0.12	0.48	0.52	0.50	**0.60**
WT12	-0.20	-0.14	0.14	0.40	0.48	**0.50**
WT13	0.16	-0.08	0.48	0.48	**0.56**	0.54
Average	0.13	0.10	0.22	0.25	0.29	0.28

ber of queries improved over the KLD baseline n^+ and the number of queries scoring lower than the KLD baseline n^- divided by the total number of queries $|Q|$. The comparison of the robustness indices in Table 1.7 shows that CPE is, on average, the most robust model.

$$RI = \frac{n^+ - n^-}{|Q|} \tag{1.23}$$

1.6.4. PROXIMITY HYPOTHESIS

We hypothesized that the usefulness of term proximity for ranking documents is not bound to a maximum word distance. To test this, we retrieved results with the CPE model on the test collections, varying the maximum span in the range $10, 20, \ldots, 1000$. In Figure 1.4, the results are shown for the TREC sets and the Web Track sets, with a MAP score that was normalized by dividing by the highest MAP for the sets, so that 1 is the maximum value. For Web collections, having no constraint on the distance used maximizes results. The results for the TREC sets reaches is optimal when using co-occurrences within 140 words, and stabilizes to a level approximately 0.1% lower over unlimited distance. Inspection revealed that this decline beyond 140 words is mostly caused by documents from the AP and WSJ sections of the TREC collections, therefore it may be a collection specific characteristic. The expected gain of finding an optimum maximum span opposed to simply using all co-occurrences will be minimal, if any. We therefore conclude that using all co-occurrences despite of their distance in text is a good principle, provided that the score of co-occurrences correctly depends on their distance in text.

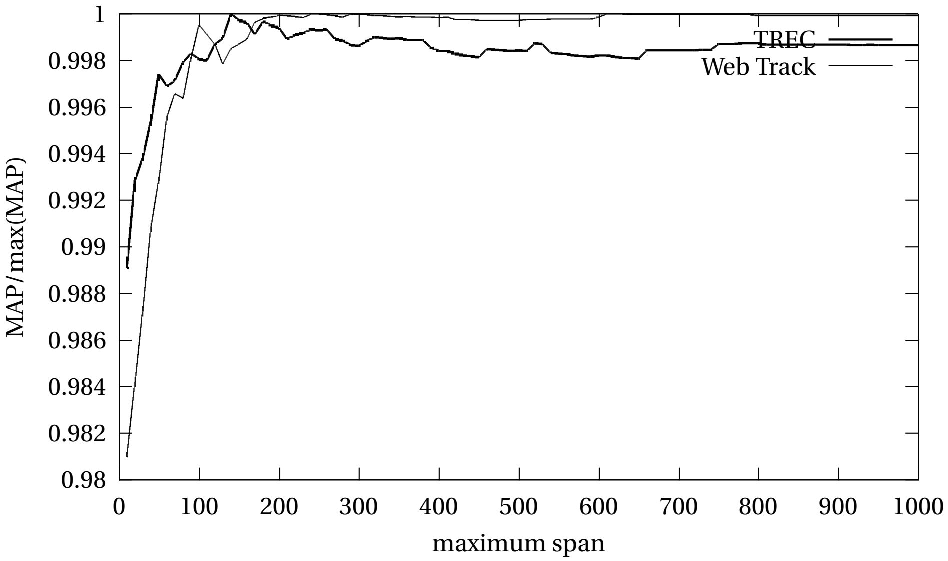


Figure 1.4: A comparison of MAP obtained for the TREC sets and Web Track sets, with a varying limit on maximum span used to match occurrences. The MAP scored per maximum span was divided by maximum MAP to normalize scores to a maximum of 1.

1.6.5. QUALITATIVE ANALYSIS

The results were analyzed in more detail to gain insight into how the distance between occurring query terms affects the document scores, what caused relevant documents to be negatively affected, and how term combinations that have an inversely proportional relation to being relevant affect the ranking. The intended effect of a proximity model is illustrated using two cases of documents being re-ranked. On topic 365 of TREC-7: “El Nino”, KLD resulted in 0.73 on average precision, ranking the irrelevant document FBIS4-11397 in 10th position. CPE ranked this specific document to the 43rd position, resulting in an almost perfect average precision of 0.96 for this topic. The fragment (from document FBIS4-11397) shows the part of this irrelevant document where “El” and “Nino” co-occur further apart, using the words “El” and “Nino” in a different meaning:

> “Four other members of the gang that killed General Julio Nino Rios, former director of the defunct Peruvian Republican Guard. (Lima EXPRESO in Spanish 4 Apr 94 p A17). Seven Tupac Amaru Revolutionary Movement, MRTA, ‘terrorists’ were presented to the press at the Fifth Infantry Division unit in El Milagro District, in the Amazon province of Bagua.”

For TREC-7 topic 397: “Automobile Recalls”, CPE improved average precision from 0.32 to 0.40. The next fragment (from document LA021390-0016), which was ranked #1 by KLD, discusses *safety seats* being recalled, not automobiles:

> “Parents often breathe a sigh of relief once they buckle their toddlers into automobile safety seats. That might be a false sense of security. Since 1968, there have been 39 recalls of various models of child safety seats.”

CPE re-ranked this document to 4th position, below three relevant documents. Both fragments illustrate that if more intermediate words are present, it becomes more likely that the relation between co-occurring query terms is diverted or changed.

Queries that were negatively affected by the proximity model were also inspected. Among the worst performing queries was TREC 1 topic 61:“Israeli Role in Iran-Contra Affair”. For some queries, the score of one vital term, in this case “Israeli”, cannot compete with the much higher proximity score of the frequently adjacently occurring “Iran Contra Affair”. Another bad case for proximity models is TREC 7 topic 377:“Cigar Smoking”, for which the distance between terms is inversely related to relevance. In this case, it is partly a side effect of stemming that causes smoke, smoker and smoking to be converted to the same stem, thus promoting documents containing “cigar smoke” or “cigar smoker”. Specifically inspecting term combinations with an inverse distance-relevance relationship, it appears that these occur mostly in queries with more than two terms. The last example is TREC 6 topic 350:“Health and Computer Terminals”, which requests information about the hazardous effects to individuals who make daily use of computer terminals. Adjacent occurrences of “computer health” and “health terminals” all appear in non-relevant documents. For this topic only one of these six documents was promoted into the top-1000 documents by CPE, specifically to position 429. The occurrences with an opposite proximity-relevance relationship are possibly restrained by the absence of other query terms which are less likely to co-occur in irrelevant documents, reducing the negative impact of these “opposites”. Although this side-effect was not studied thoroughly, on this particular topic CPE improved average precision by approx. 20% over the KLD baseline, illustrating that impact of irrelevant adjacent co-occurrences can be conditioned by the absence of other terms.

1.7. CONCLUSION

In this paper, we studied the use of term proximity information for ranking documents according to their estimated relevance, focusing on the question how the distance between co-occurring query terms may influence relevance weights associated to these terms. We first studied how the distance between query terms in documents is related to their likelihood to appear in a relevant document, using data from existing test collections. The insights from this preliminary study were used to design a simple yet effective proximity model (called Cumulative Proximity Expansions or CPE), that aggregates the estimated impact of query term co-occurrences. This impact is solely derived from the distance between their respective occurrences, motivated by the intuition that the distance between terms corresponds to the number of intervening terms that each may have modified the semantic relation between these terms.

CPE distinguishes itself from the state-of-the-art proximity models by requiring no additional parameters to be tuned, and no collection wide co-occurrence statistics. CPE is thus easily implemented, by modifying only the scoring function on any inverted index that contains positional posting lists for unigrams. We optimized retrieval speed by counting the co-occurrences of all term combinations in a single pass, which we show to be feasible even in the case of the very long queries of TREC-4. The runtime performance is close to that of its main competitor SDM, and more feasible in practice than FDM, should we encounter long queries.

We empirically compared the retrieval effectiveness between our proposed models and four different state-of-the-art baseline proximity models, over seven TREC ad-hoc and four TREC Web Track collections. Of all models, the CPE model is the most robust (in terms of the Robustness Index), while, on average, retrieval effectiveness is comparable to FDM and outperforms MinDist, PLM and SDM. The CPES variant uses a query's stop words in the term combinations considered, and outperforms the other models on the Web collections. This variant is however less robust on the ad-hoc collections, where the stop words in queries do not always help predict the documents that are most relevant.

In previous research, the use of proximity information is often limited to selected term combinations and within a limited word distance. We hypothesized that the likelihood for query terms to co-occur in a relevant document diminishes with the distance between these terms, but that the evidence from co-occurrence should not be restricted to a short window size only. Our experimental results indeed confirm the results of (a large number of) previous studies, that nearby co-occurrences of query terms provide strong evidence about a document's relevance. Scoring distant co-occurrences does however lead to further improvements in effectiveness. Although we have observed that proximity can be counter-effective in special cases of query term combinations, we have found that, generally, using all term combinations outperforms other models and provides more robust results. Nevertheless, an interesting future direction of research would analyze for which queries or term combinations proximity models are likely to improve results. More insights to predict the important term combinations (as well as those to ignore) may alleviate the negative effects of proximity as well as improve the runtime performance of the system.

II

Text and temporal proximity for news filtering

2

ONLINE NEWS TRACKING FOR AD-HOC INFORMATION NEEDS

Following online news about a specific event can be a difficult task as new information is often scattered across web pages. In such cases, an up-to-date summary of the event would help to inform users and allow them to navigate to articles that are likely to contain relevant and novel details. We propose a three-step approach to online news tracking for ad-hoc information needs. First, we continuously cluster the titles of all incoming news articles. Then, we select the clusters that best fit a user's ad-hoc information need and identify salient sentences. Finally, we select sentences for the summary based on novelty and relevance to the information seen, without requiring an a-priori model of events of interest. We evaluate this approach using the 2013 TREC Temporal Summarization test set and show that compared to existing systems our approach retrieves news facts with significantly higher F-measure and Latency-Discounted Expected Gain.

Keywords: *Information Filtering · Clustering · Multi-document summarization*

The contents of this chapter has been published as J.B.P. Vuurens, A.P. de Vries, R. Blanco and P. Mika, Online News Tracking for Ad-Hoc Information Needs. In Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, pages 1047–1048, ACM, 2015.

2.1. INTRODUCTION

Internet users are turning more frequently to online news as a replacement for traditional media sources such as newspapers or television shows. Still, discovering news events online and following them as they develop can be a difficult task. Although the Web offers a seemingly large and diverse set of information sources ranging from highly curated professional content to social media, in practice most sources base their stories on previously published works and add a much more limited set of new information. Thus users often end up spending significant amount of effort re-reading the same parts of a story before finding relevant and novel information. Most recently, the TREC Temporal Summarization track[1] have taken up this challenge, promoting research in the area of *online* news summarization, i.e. focusing on developing news, as opposed to archival news. Online summarization is a crucial aspect of real-world products such as online live streams for natural disasters, product launches, financial or political events, breaking news notifications on mobile devices and topical daily news summaries like Yahoo! news digest[2].

In this study, we propose a novel approach for the temporal summarization of news. Our approach works in an online fashion and provides previously unseen information related to a predefined ad-hoc information need, expressed as a user query. Contributions of this work are:

- the use of a specifically designed clustering approach to detect news that is supported by multiple online providers,
- the online selection of the best sentences according to a specifically tailored relevance model over recently seen information, that allows the retrieval of unanticipated information by adapting to information recently seen instead of requiring an a-priori model of events of interest,
- an approach that requires no manual intervention and contains a small number of parameters that can be tuned in straightforward fashion.

We evaluate our approach using the 2013 TREC Temporal Summarization test set. In these experiments, our approach significantly outperformed the top performing systems on both F-measure and Latency-Discounted Expected Gain. To facilitate further research in this area, we also publish our implementation of the described model, the results of empirical experiments and the annotated ground truth[3].

The remainder of this paper is structured as follows: Section 2 discusses related work in the area of temporal summarization of online news information and the necessary prerequisites. In Section 3, we present our approach to extract sentences containing news facts from an online stream of news articles. In Section 4, we describe the implementation, test set used for the empirical evaluation, and how the data in the collection was processed. In Section 5, we present the results of the empirical evaluation, and analyze parameter sensitivity. The conclusions are presented in Section 6.

[1] `http://www.trec-ts.org/`

[2] `https://mobile.yahoo.com/newsdigest`

[3] `http://newstrackerpaper.github.io/`

2.2. RELATED WORK

2.2.1. NEWS TRACKING AND SUMMARIZATION

The task of detecting events can be automated using information about the events published online. For this purpose, the Topic Detection and Tracking (TDT) program was initiated to discuss applications and techniques for detecting and tracking events that occur in real-time and the infrastructure to support common evaluations of component technologies. The *tracking of news* involves the online identification of stories that discuss a targeted event, which needs to begin as soon as a only a few training documents have become available to model a real world setting. For this, Allan et al. present an information filtering approach, in which a tf-idf vector made from training documents is used as a query to match only documents that exceed a similarity threshold. In one experiment, “surprising” (previously rarely seen) words were used for tracking events, but they found that these words do not provide a broad enough coverage to capture all stories on the event and that many of these “surprising” words are useless for retrieval. They also found that a query based on initial training documents does not allow to track stories when the discussion of an event changes over time. For some queries at least, results were improved by using a tracking model that adapts the query based on new information seen, similar to the notion of pseudo-relevance feedback (Allan et al., 1998).

The temporal summary of news stories can help a person monitor changes in the coverage of news stories over time, which are typically very redundant and increase the effort required to identify genuinely new information (Gabrilovich et al., 2004). The core technique of temporal summarization is to summarize multiple texts by extracting salient sentences. Regarding measures of salience that can be used to choose the best sentences for news summarization, the literature provides no clear consensus. Two general criteria to select the best candidates sentences are the most *useful* and *novel* sentences, i.e. related to the topic and non-redundant. Techniques that use these criteria for instance consider the words in the sentences, look for cue words and phrases, consider features such as sentence length and the case of words, or compare patterns of relationships between sentences. Often, these approaches use statistics from the corpus itself to decide on the importance of sentences, and some leverage existing training sets of summaries to learn the properties of a summary (Allan et al., 2001). Candidate sentences can subsequently be ranked based on estimated importance (e.g. Erkan and Radev, 2004; Radev et al., 2004; Tran et al., 2015). Some work has focused more specifically on the summarization of news in an online setting. Radev et al. (2005) presented a news delivery and summarization system "News In Essence", that supported retrieval of news related to a document that the user provided. Gabrilovich et al. (2004) present a methodology for filtering news stories based on novelty, by selecting the articles that are most different to those already read. This work also focuses on summarization in an online setting.

The salience of sentences is more easy to determine in retrospect than for online systems (Yang et al., 1998). In retrospect, there is more information to compare possible solutions based on size and the coverage of the possible relevant facts over the stream of redundant information. Erkan and Radev (2004) argue that sentences that are similar to many of the other sentences in a cluster are more central (or salient) to the topic, and propose an algorithm with resemblances the HITS algorithm that uses similarity edges instead of hyperlinks to estimate the salience of sentences. Yu and Hatzivassiloglou (2003) present an approach to detect opinions in contrast to factual information with very high precision and recall, by using a fairly straightforward Bayesian classifier. In recent work, Tran et al. (2015) used this classification approach in reverse to select headlines containing factual news. They further refined their detection of salient headlines by assuming that a higher *spread* indicates more important news, and that the relatedness to subsequent events indicates *influential* news, but it appears their approach is more specific for summarization in retrospect.

The clustering of information that discusses the same topic can be useful for several purposes related to temporal summarization. Some studies use the heuristic that the most similar sentences tend to be salient, which can be detected using clustering (Erkan and Radev, 2004; Fiscus and Doddington, 2002; Leban et al., 2014). Clustering has also been used to extract concise information from redundant sources (Allan et al., 2000; Radev et al., 2004). Allan et al. (2000) experimented with different variants and obtained better results using single linkage clustering with cosine similarity. In single linkage clustering, every data point is assigned to its nearest neighbor, in accordance to the k-Nearest Neighbor (kNN) decision rule described by Cover and Hart (1967) for classification. Cover and Hart show that for the classification of n-samples there exists no $k \neq 1$ with a lower probability of error than $k = 1$ against all distributions. A known problem for finding nearest neighbors in large datasets is that the required number of computations increases quadratically (Petrović et al., 2010).

The main contribution of this work is a novel approach to select the most salient sentences in a news stream, by leveraging the redundancy that is typical between news articles that discuss the same event. We introduce a variant of kNN clustering called 3-NN, which differs from existing work by forming clusters around a minimum of three sentences that are in each others' $k = 3$ sets of nearest neighbors and published by different news agents. For the online summarization of these salient sentences, we use an adaptive approach that resembles that of Allan et al. (1998), but is rather based on recently seen salient sentences to limit the selection of sentences to the most relevant according to the most recent news. The complete system we used to evaluate these efforts can be viewed as a hybrid combination of techniques for query based online news tracking and summarization, adapted from Allan et al. (1998, 2001). Our approach differs by a stronger emphasis on novelty of information emitted (e.g. Gabrilovich et al., 2004). Hereto, we estimate the amount of previously unseen information to use only sentences that are likely to contain novel information.

2.2.2. TREC Temporal Summarization

In recent years, TREC stimulated research on online summarization of news related to a specific topic or query, by initiating the Temporal Summarization track. The PRIS team participated with a manual system in the 2013 edition of the TS track and obtained the highest Expected Gain. They use hierarchical Latent Dirichlet Allocation on documents describing similar events as the topic to mine ten subtopic descriptions per TREC topic. From the generated topic descriptions they manually selected the keywords that describe each topic best. The sentences that are most similar to the selected keywords of a topic are selected as output (Zhang et al., 2013). In the same track edition, ICTNET obtained the highest F-measure of all participants. A list of relevant words is learned from training documents, which are then matched to the sentences of documents that contain all query terms in the title. A matching sentence is then compared to previously emitted sentences, and removed if the similarity exceeds a threshold (Liu et al., 2013). These participants provided the best performing runs, out of 27 submitted for this task, and we will compare our results to the results of these systems in the evaluation. These query based approaches dominantly make use of a model crafted over similar events, e.g. other earthquakes or train crashes documented on Wikipedia. These approaches are optimized for retrieving the same, often reported types of information about common types of events, but may fail when the type of the event is not known or the type of information is not typical for the type of event. The proposed method uses only a single query to represent the event and does not require further training data.

2.3. DESIGN

News facts can be obtained from several sources on the Web, e.g. online news sites, blogs, social media, Wikipedia. One advantage over traditional broadcast news is that online news facilitates easy access to additional information. However, manually tracking relevant and novel news facts online is rendered inefficient by the high redundancy between multiple sources that discuss more or less the same information. This research focuses therefore on the automated extraction of relevant and novel news facts for ad-hoc information needs, allowing to push newly published facts to a user the instant they are published; or, alternatively, to present the user a summary of the most important news facts over a timeline. Additionally, presenting the most important news facts on a timeline may also be useful to help keep update knowledge bases up-to-date, such as Wikipedia or the knowledge graphs used by search engine companies. From an end-user perspective, we consider it important that a high percentage of results is on-topic, and therefore this study uses news articles as the sole source. We expect their content to be mostly factually correct, timely, and presented in an accessible form (Radev et al., 2005). Events that are of interest to many people are naturally reported in different news articles, from different sources (Allan et al., 1998; Chieu and Lee, 2004). In our approach, we leverage the redundancy between news articles, clustering sentences that are likely to discuss the same news facts to select salient sentences and to avoid biased information (Leban et al., 2014). Eventually, our work may serve as a baseline to evaluate approaches that also consider alternative sources like social media.

In this Section, we describe a new approach to extract sentences from an online stream of published news articles that are related to a user's ad-hoc query. We operate in a strict online setting, processing the articles one at a time as they arrive. The remainder of this Section first discusses observed characteristics for factual news. We outline the process that is proposed to extract sentences containing news facts from a stream of online news articles, followed by a detailed discussion of each step in this process.

2.3.1. NEWS EXTRACTION PROCESS

We first outline the proposed method for the online tracking of ad-hoc user needs in a stream of news articles, which consists of three steps: *route*, *identify salient sentences* and *summarize*. The key method underpinning our approach is a clustering method that takes care of both the routing and the identification of salient sentences. In the first step, a single graph is maintained in which all news articles are clustered, and 'query matching clusters' are *routed* to a query specific module to identify salient sentences. In this second step, per query that is being tracked, we cluster the contents of clusters that match that query to *identify* the most central sentences, which we consider the most salient ones. In the third step, per query that is being tracked we *summarize* the salient information by qualifying only the most novel and useful sentences from the current document.

Routing

The first step of the outlined process identifies clusters of news articles by several news agents that share information, and route 'query matching clusters' to the designated identification and summarization process that is executed per query. Here, we define *query matching clusters* as the clusters that contain at least one news article that matches that query; in this study, an article matches a query when all query terms appear in its title. This section first gives a rationale for the features used to assign a document's nearest neighbors, and then describes the clustering method in detail.

To estimate which news articles are likely to discuss the same event, we use the similarity of the titles and the proximity of the publication times. The use of titles is motivated by the observation of Tran et al. (2015), that news article titles are often short sentence abstracts of the news contained, to allow readers to gain a quick overview of the news based on titles and to invite them to read the full article if it is of interest to them. Additionally, titles contain less words than entire documents, and so the collection of news article titles can be fitted into the memory of a single computer, allowing to process the data without the need to partition it. The latter is primarily a practical argument when developing an online news summarization approach. The use of proximity in publication times is motivated by the observation that stories about the same event often occur in proximate time, most particularly for unexpected events where the news media exhibit strong interest in a story (Allan et al., 1998; Yang et al., 1998).

We introduce a *3-NN* streaming variant of k-Nearest Neighbor clustering, that assigns directed edges to each article's three nearest neighbors while not allowing nearest neighbor links within the same web domain. We use an online algorithm to detect newly formed clusters as 2-degenerate cores, according to the theory of k-degenerate graphs (Meladianos et al., 2015). These 2-degenerate cores identify the most central information based on similarity in content, proximity in publication time and support by multiple news agents. The selected news is therefore is more likely to be factual, correct and important.

In 3-NN, a new 2-degenerate core is formed only when the arriving node is part of a bi-directional loop of nodes that is currently not clustered. Multiple bi-directional loops that are connected by a single bi-directional edge are considered to be separate clusters. Nodes that are not part of a 2-degenerate core are still assigned to a cluster if their majority of nearest neighbors is a member of the same cluster. Figure 2.2 illustrates the online process that takes place upon the arrival of new articles (that correspond to nodes in the graph), when clusters are formed, expanded or disbanded. Edges in the graph point to one of a node's k-nearest neighbors, labeled with the similarity between the nodes. Dashed arrows indicate the similarity between new arriving nodes and existing nodes.

Considering the example of Figure 2.1a in more detail, nodes A and B are not clustered because there is no evidence that the majority of both nodes nearest neighbors must belong to the same cluster (C and D could belong to different clusters). When a new node F arrives (Figure 2.1b), it is compared to the existing nodes, to assign its three nearest neighbors. Since F is more similar to B than B's currently weakest nearest neighbor E, an edge from B to F will replace the edge from B to E. After F has been added (Figure 2.1c), nodes A, B and F form a bi-directional loop. For this particular situation, we can deduce that A, B and F must have their majority of nearest neighbors in the same cluster, and therefore they form a cluster. We will refer to the nodes that form a bidirectional loop to establish a cluster as its *core nodes*. In Figure 2.1d, E is added to the cluster consisting of A, B, F, because its majority of nearest neighbors connect to that cluster. In Figure 2.1e, when a new node G arrives that is more similar to A than its weakest of nearest neighbors F is, the edge from A to F will be replaced by an edge from A to G. With this change, the bi-directional loop from which we deduced the existence of a cluster A, B, F, is now gone. Therefore, in Figure 2.1f, there is no more cluster.

The nearest neighbors of a given title or sentence are found by computing the similarity to all other titles or sentences. The similarity between two sentences s_i and s_j is scored using Equation 2.1, which combines the cosine similarity between the binary vector representation of the two sentences with the difference in publication time, in accordance to the observations by Yang et al. (1998). Equation 2.2 estimates the temporal proximity of two publications, $\tau \in [0,1]$, as the absolute time between the publication times of $s_i.t$ and $s_j.t$, truncated by a constant maximum period T. Equation 2.3, δ is a function that guarantees that assigned nearest neighbors are published within a time span with duration T, and originate from a different source domain ($s_i.d \neq s_j.d$).

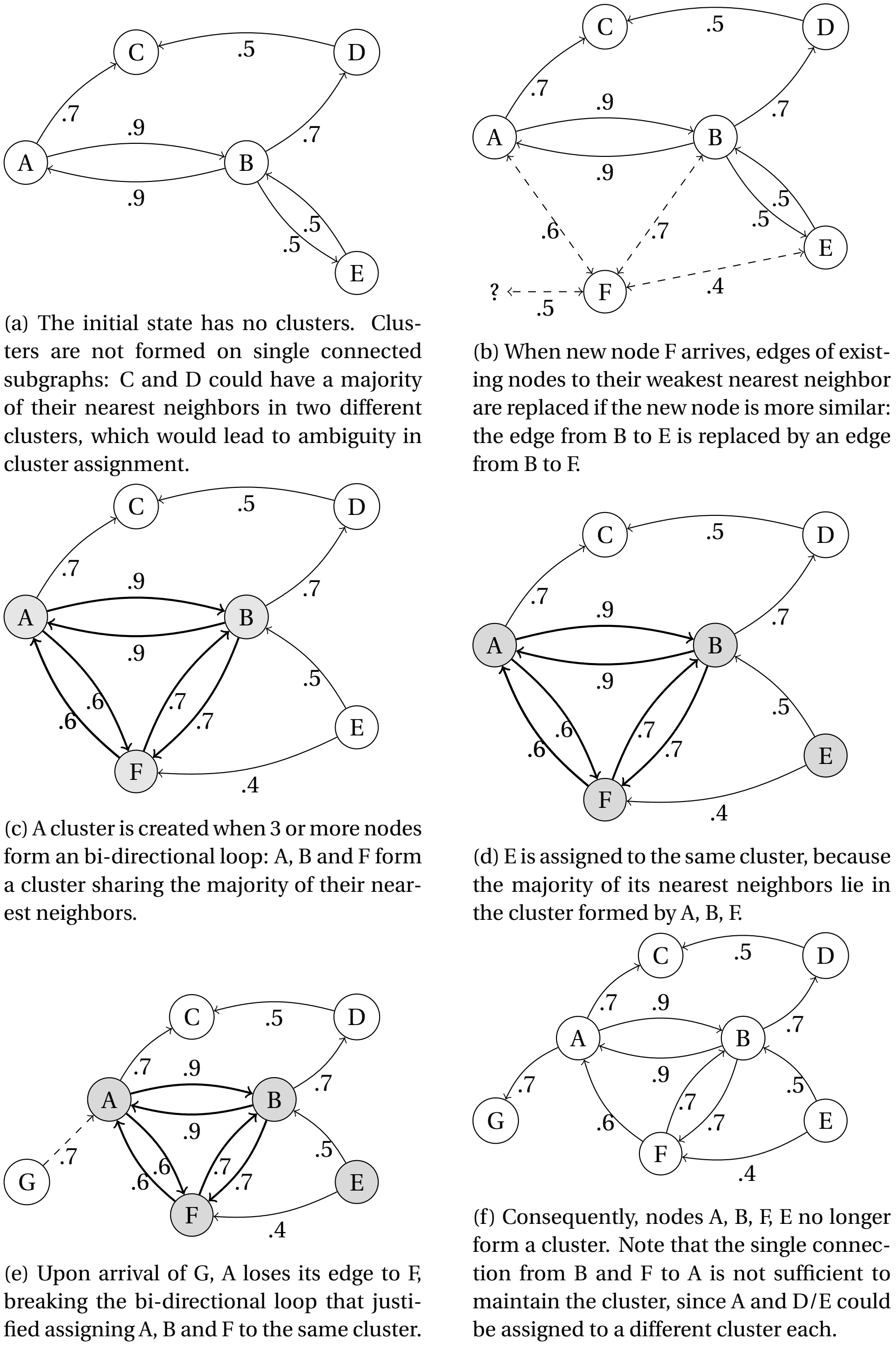
$$sim(s_i, s_j) = cos\left(s_i, s_j\right) \cdot \tau\left(s_i, s_j\right) \cdot \delta\left(s_i, s_j\right) \tag{2.1}$$

$$\tau(s_i, s_j) = 1 - \frac{|s_i.t - s_j.t|}{T} \tag{2.2}$$

$$\delta(s_i, s_j) = \begin{cases} 0, & \text{if } |s_i.t - s_j.t| > T \text{ or } s_i.d = s_j.d \\ 1, & \text{otherwise} \end{cases} \tag{2.3}$$

IDENTIFICATION OF SALIENT SENTENCES

For each tracked query, we identify salient sentences in a separate graph. The routing will result in forwarding the clusters that match a query to the corresponding 'sentence graph', to which a node is added for every sentence in the query matching clusters. For sentences we follow an analogous rationale as for titles; salient sentences are likely to be published in proximate time and share information and are thus likely to be clustered together, and we therefore cluster the sentences according to the same 3-NN heuristics as described above. Within the clusters of such a 'sentence graph', the core nodes are the most central sentences and thus in this study regarded as the most salient. Operating in an online setting, we only consider sentences from the current document as *candidate sentences* for the news summary. However, if candidate sentences are clustered, their



(a) The initial state has no clusters. Clusters are not formed on single connected subgraphs: C and D could have a majority of their nearest neighbors in two different clusters, which would lead to ambiguity in cluster assignment.

(b) When new node F arrives, edges of existing nodes to their weakest nearest neighbor are replaced if the new node is more similar: the edge from B to E is replaced by an edge from B to F.

(c) A cluster is created when 3 or more nodes form an bi-directional loop: A, B and F form a cluster sharing the majority of their nearest neighbors.

(d) E is assigned to the same cluster, because the majority of its nearest neighbors lie in the cluster formed by A, B, F.

(e) Upon arrival of G, A loses its edge to F, breaking the bi-directional loop that justified assigning A, B and F to the same cluster.

(f) Consequently, nodes A, B, F, E no longer form a cluster. Note that the single connection from B and F to A is not sufficient to maintain the cluster, since A and D/E could be assigned to a different cluster each.

Figure 2.2: Explaining when clusters are created and broken using the nearest neighbor heuristic, $K = 3$, with the requirement that nodes are only clustered when they are members of a 2-degenerate core or when their majority of nearest neighbors is a member the same cluster.

entire cluster will be passed to the summarization step, since the cluster provides part of the context needed to qualify (future) candidate sentences.

Summarization

In general, for an optimal summary of news we should select sentences that are the most useful and novel, i.e. related to the topic and non-redundant with other sentences in the summary (Allan et al., 2001). In this step, we qualify which candidate sentence(s) are added to the news summary. Obviously, this is easier to optimize in retrospect than in an online setting, since we must decide whether or not to use a sentence without knowledge of what is yet to come. Operating in an online setting, we only consider sentences from the current document to use in the summary. Once the decision has been made to add a sentence to the summary, this cannot be reversed if the original sentence is removed from the cluster when a new sentence arrives. For the qualification we formulate a set of heuristics to select useful and novel sentences.

Erkan and Radev (2004) hypothesize that sentences that are similar to many other sentences in the cluster are more salient to the topic. In our 3-NN clustering, the core nodes are thus likely to be the most salient sentences. Initially, we expand clusters by adding non-core nodes that have a majority of nearest neighbors in one cluster. However, nodes can be assigned to a non-related cluster in the absence of closely related content. A directed path from a core node to another cluster member is likely to identify closely related content. To reduce the risk of using off-topic sentences, we apply a variant of graph peeling (Abello and Queyroi, 2013) by removing nodes to which there exists no directed path from a core node. In the remainder of this section when we refer to cluster members we only consider the cluster members for which a directed path exists from a core node.

In our approach, a redundant stream of news articles is aggregated into a concise summary by selecting only sentences that are most relevant to the most recent developments for the topic. Without the use of training documents, we obtain a model of the most important information from the news stream, however, what information is important for a topic can change over time (Allan et al., 1998). Yang et al. (1998) observed that a time gap between bursts of topically similar stories is often an indication of different events, suggesting a need for monitoring cluster evolution over time and a possible benefit from using a time window for event scoping. If significant shifts in vocabulary indicate stories that report a novel event, this motivates the use of an adaptive model that allows to identify novel events. Analogous to Bates (1989), we propose an unsupervised ‘berry-picking’ approach that estimates relevance at some point in time based on the information seen in a window over the prior h hours, to compare the estimated relevance of the candidate sentences to sentences already summarized, and selectively qualify only candidate sentences that rank among the top-r sentences. The rationale for this berry-picking approach is that news topics tend to evolve over several subtopics; consider for example a crime happening, the police investigation, a suspect being arrested, etc. Some subtopics are repeatedly reported over a longer period, while others are mentioned only briefly. We construct a relevance model per news topic (a current ‘event profile’), which is initially seeded with the user’s query terms. The model is continuously expanded with the core node sentences from all query matching clusters to limit the risk of adding off-topic information. An adaptive relevance model is obtained at time t by removing sentences that were published before $t-h$ hours, allowing to shift the notion of relevance to recently seen information. In the event the relevance model contains no sentences published after $t-h$, the relevance model returns to the original query terms. For ranking, we express the relevance model for the news topic at a given a point in time as a word vector, where the frequency of each word is the number of sentences it appeared in over the last h hours. The candidate sentences of the latest arriving document are then ranked among the sentences currently in the summary, using the cosine similarity between each sentence and the relevance vector. Candidate sentences that are ranked outside the top-r are disqualified for use in the summary.

New sentences that do not share any words with information already seen can disorient the reader, being possibly off-topic as well. To reduce topical drift and improve readability of the timeline created, we require qualified sentences to contain at least one of the query terms and two words that appear jointly in either the query or in a sentence already used in the summary. Formally, in Equation 2.4, we define $WC(s)$ as the collection of all combinations of words (w_1, w_2) that appear in sentence s, and $QWC(s,q)$ as the subset of $WC(s)$ in which at least one of the words appears in the query q. In Equation 2.6, we define K as the collection of all word combinations containing at least one query term that was previously seen in either the query q or one of the sentences in the summary S. Finally, in Equation 2.7 is the constraint that at least on of the word combinations in the candidate sentence c must be in $K(S,Q)$. This simple requirement effectively filters out the (unrelated) sentences that still form clusters, such as navigational elements or links to other news stories.

$$WC(s) = \{(w_1, w_2) \mid w_1 \in s \wedge w_2 \in s \wedge w_1 < w_2\} \tag{2.4}$$

$$QWC(s,q) = \left\{(w_1, w_2) \in WC(s) \mid w_1 \in q \vee w_2 \in q\right\} \tag{2.5}$$

$$K(S,q) = \cup_{s \in S} QWC(s,q) \cup WC(q) \tag{2.6}$$

$$K(S,q) \cap WC(c) \neq \emptyset \tag{2.7}$$

Additionally, qualified sentences must add information that is not previously seen and is supported by another source. Previously unseen information could be simply measured by the number of previously unseen unigrams. Alternatively, the amount of information shared by sentences can be estimated by the number of two-word combinations that appear jointly in both sentences, which is possibly less affected by noise and will be used unless stated otherwise. Formally, in Equation 2.8, we define the set of possible sentences that can provide support for word combinations $SUP(CL,c)$, as the sentences s in cluster CL that are published on a news site $s.d$ that is different from the news site of the candidate sentence $c.d$. In Equation 2.9, we estimate the amount of novelty N as the number of two-word combinations that appear in both the candidate sentence c and a sentence on a different news site, but not in one of the sentences that was used in summary S. In Equation 2.10, we set a threshold based on the number of possible word combinations that contains at least one non-query term. We use a parameter $n \in [0,1]$ to control the fraction of two-term combinations that must be novel and supported to qualify a sentence to use in the summary.

$$SUP(CL, c) = \{s \in CL | s.d \neq c.d\} \tag{2.8}$$

$$N(c, CL, S) = | \cup_{s \in SUP(CL,c)} WC(s) \cap WC(c) - \cup_{s \in S} WC(s)| \tag{2.9}$$

$$N(c, CL, S) >= (|c - q|) \cdot (|c| - 1) \cdot n \tag{2.10}$$

2.4. EXPERIMENT

2.4.1. FEASIBILITY OF ONLINE KNN CLUSTERING

Clustering all news articles using the nearest neighbor heuristic, requires the computation of similarity of each news article against all others. For incremental online clustering, the number of required comparisons can be reduced by using a criterion to remove nodes and clusters that are outdated, by Aggarwal and Philip (2010) referred to as 'cluster death'. Since in this approach a zero score is assigned between sentences with a publication time more than T away (see Equation 2.3), we can do so with a high probability of not affecting clustering results. Since the news sentences of such a limited period of time fits into memory, we do not require an approximation such as Latent Semantic Hashing to partition the data. Additionally, we use in-memory posting lists on the words that appear in sentences, so that we do not compare sentences that have no word in common. In practice, this results in an algorithm of order $n \cdot log(n)$. Figure 2.3 shows the clustering efficiency over a stream of news articles in the KBA corpus, for 21 days from 2011-11-06, that was clustered on a standard laptop, in approx. 100 seconds. On the left-hand side of the graph, we observe that the clustering speed slows down slightly when more articles are in memory. The vertical drops in the graph are the result of removing 'expired' articles as discussed above. This graph shows online processing of all published news titles is feasible using the proposed clustering approach.

In the proposed 3-NN clustering method, nodes that do not have 2 nearest neighbors in the same cluster correspond to the 'outliers' of Aggarwal and Philip (2010) and remain un-clustered. In our experiments using sentences of news articles, on average 20% is un-clustered at any given time.

In theory, a chain of nearest neighbors could span a period greater than T, and, although unlikely, cluster assignment could be affected over a larger time span. To allow for such anomalies, at the end of each day we prune sentences older than $T + 1$ days, except for clustered sentences which are not pruned until all its members are older than $T + 1$ days. For the 2013 KBA Streaming corpus, we compared the clustering results of a pruned run to a run that does not prune the articles from memory, and confirm that the clustering is not affected by removing 'expired' items.

2.4.2. EVALUATION

To evaluate our approach, we used the test collection from the Sequential Update Summarization task at the 2013 TREC Temporal Summarization track, and compare effectiveness against the two best performing systems. For this track, the 2013 KBA Streaming corpus was used, in which the documents are already parsed into sentences by the organizers, and the sentence numbers are being referred to from the existing ground truth set. The task is to retrieve a list of timestamped extracted sentences (referred to

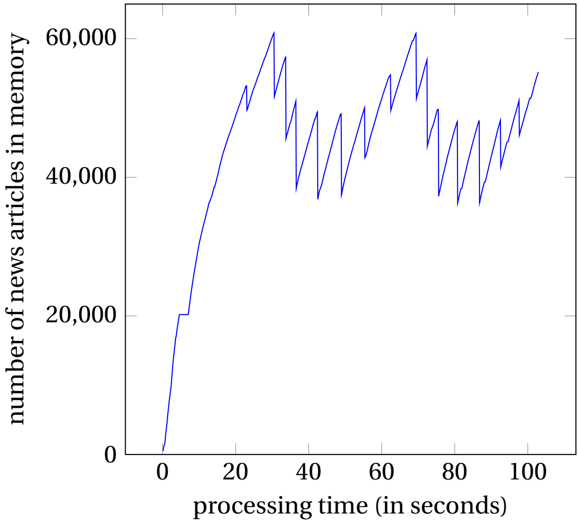


Figure 2.3: during the clustering of a stream of 3 weeks, the number of news articles in memory over time

as *updates*), for a set of 9 topics that contain a query referring to a news event. The effectiveness of a system is measured using a set of gold standard updates (referred to as *nuggets*), that were extracted from Wikipedia event pages and timestamped according to the revision history of the page. The TREC participants submitted a list of updates, from which a pool of 3,268 updates was manually compared to the 1275 identified nuggets for 9 topics, resulting in 2,416 matches between updates and nuggets and 2,142 updates that do not match a nugget (one update can match multiple nuggets).

Of the sentences returned in our experiments we found that an insufficient number has been annotated by TREC to obtain reliable metrics (see Section 2.5.4 for the empirical data and discussion). As a resolution, we manually annotated all missing sentences *against the existing nuggets*. During the annotation, to the best of our ability we matched retrieved results to those scored by TREC annotators to score our results consistently. Occasionally, we encountered updates that seemed very relevant but could not be matched to any nugget. Since we assume that no nuggets were added in the process of scoring the updates for participating TREC systems, we did not add any nuggets to the ground truth. Adding the updates to the pool without a matching nugget is equivalent to scoring these as irrelevant.

2.4.3. DATA PROCESSING AND CLEANING

In an exploratory phase we used a crawl of online news articles over the first part of 2014 for construction and training of the system. For this crawl, we extracted a list of domains that are referenced on the Wikipedia Current Event Portal between January 1st 2013 and September 1st 2014, from the WikiTimes portal (Tran and Alrifai, 2014). We removed all domains from Asia, Africa, non-English and non-news domains, resulting in 141 domains. For the evaluation on the KBA Streaming corpus, we use the same system and consider only news articles from the described domains.

The KBA Streaming corpus contains the original HTML source of the documents and sentences that were extracted by the organizers. This extraction was done using rudimentary heuristics, which in the absence of periods occasionally produced sentences of several hundreds of words that for instance include entire paragraphs, tables or navigational labels. Since our approach specifically depends on the quality of title clusters, we extracted the actual document titles from within the HTML title tags, stripped non-news elements (e.g. categories and news paper names) using a manually constructed list of general and domain specific regular expressions (e.g. truncating titles after the a dash and removing the word TIME if this was the last word in a title from the `time.com` domain). These actual titles are used for clustering the articles. For a fair comparison of the proposed model to the best TREC participants, we performed a *no titles* run that emits only sentences as extracted by the TREC organizers and thus is conform to the TREC guidelines. In Section 2.5.1, we will compare the performance to a run that does allow *HTML titles* to be emitted.

For processing, all sentences were tokenized by separating tokens on non-alphanumeric characters, the tokens were lowercased, and stop words were removed, however, we did not use any stemming.

2.4.4. PARAMETER SETTINGS

The approach proposed in this paper contains several parameters: k as the number of nearest neighbors used for kNN clustering, T as a time period used to discount the difference in publication time in the similarity function (Equation 2.2), r for the rank to be obtained to qualify a sentence to use in the summary (Section 2.3.1), l for the maximum length allowed for sentences used (Equation 2.3.1), h for the time in hours used for the relevance model (Section 2.3.1), and n to control the minimum amount of novel information an qualified sentence must have (Equation 2.10). In the exploration phase of this research we analyzed the effect of these parameters on seven topics that were annotated using the guidelines of the TREC TS track, and on online news that is tracked in a live demo (Vuurens et al., 2015c).

For the number of nearest neighbors, we used a fixed setting $k = 3$. By using an odd number of nearest neighbors there is no need to resolve ties. A value of $k > 1$ increases the likelihood to cluster around information that is supported by several news domains, while compared to high settings for k a low setting for k is likely to retrieve news faster and may improve recall. We leave the comparison of different values of k for future work, noting that this may be especially useful in more redundant domains like social media. For T, we have used a fixed setting $T = 3$ days throughout our study, based on the observation that it is not uncommon for news providers to post news that is more than a day old and allowing these articles to be clustered with the same content brought more promptly by other providers. Each of the remaining parameters was added to restrain the model in some respect. We observe that clustering results may vary largely dependent on parameter settings, which is possibly due to the high redundancy that is typical for news collections. The necessity of new manual annotation for each clustering outcome renders parameter training practically infeasible. Despite the variation in clustering results, the overall system performance is largely unaffected by changes in parameters.

Therefore, we use a set of default settings $r = 5$, $l = 20$, $h = 1$ and $n = 0.3$, and will show in Section 2.5.2 that the model performance is insensitive to parameter sweeps.

2.5. RESULTS

2.5.1. COMPARISON OF TEMPORAL SUMMARIZATION

For evaluation of the proposed method, we follow the guidelines of the Sequential Update Summarization task of the 2013 TREC Temporal Summarization track (Aslam et al., 2013). The effectiveness is measured using *Mean Expected Gain*, and *Mean Comprehensiveness*, which are similar to the traditional notions of respectively precision and recall in information retrieval systems, and we additionally use the *Mean Latency Discounted Expected Gain* in which the gain is discounted based on the difference between the time of the first update that matches a nugget and the time the corresponding fact was added to Wikipedia. Formally, in Equation 2.11, the gain G of an update u in a set of updates S is based on a gain function g on the nuggets n for which u is the earliest matching update as returned by the function M^{-1}. In Equation 2.13, the Mean Expected Gain MEG_v for a system is the average gain over a set of events ε, for each of which the system produced sets of updates S^e (emitted sentences), for g (Equation 2.11) a binary function is used that returns 1 if an update matches a nugget, and the total gain is normalized by the verbosity of the updates $V(u)$ (Equation 2.12), which discounts by the number of words in u that are not part of an earliest matching string for a nugget divided by the average number of words in the strings of nuggets $|words_n|$. In Equation 2.14, the Comprehensiveness C for a set of updates S for a specific event is number of matched nuggets G divided by the number of available nuggets for the event $|N|$. In Equation 2.15, the Mean Comprehensiveness MC is computed over all events ε. The Latency-Discounted Expected Gain is a variant of Equation 2.13 by using a modified function g (Equation 2.11) in which the binary relevance of matched nuggets is discounted by a monotonicaly decreasing function over the difference between the time of the earliest matching update en the time it was put on Wikipedia. For more details regarding these metrics, we refer to Aslam et al. (2013). We also report the variant of the *F-measure* that summarizes the Expected Gain and Comprehensiveness in one metric and was used as the primary metric of the 2014 Temporal Summarization track.

$$G(u,S) = \sum_{n \in M^{-1}(u,S)} g(u,n) \tag{2.11}$$

$$V(u) = 1 + \frac{|all\_words_u| - |nugget\_matching\_words_u|}{avg|words_n|} \tag{2.12}$$

$$MEG_v = \frac{1}{|\varepsilon|} \sum_{e \in \varepsilon} \left(\frac{1}{\sum_{u \in S^e} V(u)} \sum_{u \in S^e} G(u, S^e) \right) \tag{2.13}$$

$$C(S) = \frac{1}{|N|} \sum_{u \in S} G(u,S) \tag{2.14}$$

$$MC = \frac{1}{|\varepsilon|} \sum_{e \in \varepsilon} C(S^e) \tag{2.15}$$

Table 2.1: Comparison of performance using the 2013 TREC TS track against the top participants. † significant improvements over PRIS, ‡ significant improvement over ICTNET, using paired Student t-Test, 2-tailed, p < 0.05

System	Expected Gain		Latency DEG		Comprehension	F-Measure	
PRIS-cluster5	0.1491		0.1364		0.0994	0.060	
ICTNET-run2	0.1024		0.1270		0.1921	0.067	
no titles	0.2607	‡	0.3067	†‡	0.1778	0.106	†
HTML titles	0.2449	‡	0.3019	†‡	0.1901	0.107	†‡
unigram	0.2474	‡	0.2934	†‡	0.1700	0.101	†‡
IDF weighted	0.2100	‡	0.2763	†‡	0.1664	0.093	†

In Table 2.1, we compare four variants of our approach with the top participants of the Temporal Summarization Track. The "no titles" variant only uses the extracted HTML titles for clustering, but never uses these in the summary, therefore the results of this run are conform the track guidelines and comparable to other TREC participants. For the "HTML title" variant, we additionally allowed emission of the actual HTML titles which was not an option for TREC participants, the "unigram" variant is the same as "HTML titles" except that it measures new and previously seen information using unigrams instead of two-word combinations, and the "IDF weighted" variant uses the inverse document frequency obtained from Wikipedia on January 2012 (which predates the test collection) to compute the cosine similarity between sentences.

The results show that the "no titles" variant is significantly more effective than the top TREC in both F-measure and Latency-Discounted Expected Gain. Statistical significance was tested using a paired Student t-Test, 2-tailed, p < 0.05. Given the low number of topics, we also tested significance using Wilcoxon Signed-Rank test, 2-tailed, p < 0.05, which confirmed the significant improvements for all but the improvement in F-Measure of the "unigram" variant over ICTNET. At the topic level, our approach was outperformed by PRIS on topic 5 "Hurricane Isaac", and by ICTNET on topic 6 "Hurricane Sandy", using approaches that target words typically seen on the Wikipedia pages of hurricanes such as wind speeds, casualties and damage.

Compared to the "no titles" variant, the "HTML titles" variant obtains higher Comprehensiveness and a relatively higher Latency-Discounted Expected Gain. Possibly, some facts are only used in titles and some facts are introduced in titles before they are used in sentences. In Section 2.5.3, we look into the differences observed between these variants in more detail.

2.5.2. PARAMETER SENSITIVITY

To analyze parameter sensitivity, we performed parameter sweeps for the "HTML titles" variant, and plotted the results in Figure 2.4. During each sweep we changed only one parameter, using the default settings described in Section 2.4.4 for the remaining parameters.

Interestingly, we observe that the efficiency is insensitive to the size h of the time window used to estimate a relevance model of recently seen information. Possibly, news that is important is mostly reported by different agents within half an hour, explaining why the effectiveness is comparable for a window of that size. For the rank r a sentence must obtain to qualify, we expected an increase in Comprehensiveness when increasing the size, but this effect is only observed for $r < 5$. For n, which controls the minimum amount of novel information a qualified sentence must add to the summary, a low n will more greedily use sentences with a relative small amount of novel information, resulting in a classic trade-off of recall for precision. In these experiments, setting $n > 0.5$ hurts performance, possibly because sentences that contain novel information often include previously seen information.

On this particular test collection, sentence extraction by the TREC organizers occasionally resulted in large parts of content being mistaken for a sentence, to which our model is particularly sensitive. When our approach is used on a stream of correctly parsed news sentences, the maximum sentence length l could become obsolete, since the results show a higher setting of l results in slightly higher comprehensiveness and F-measure. However, these metrics do not take into account that shorter sentences improve readability, which may be preferable on mobile platforms.

In our evaluation, the difference in performance for alternate parameter settings is marginal when compared to the difference with competing systems. Therefore our default parameter settings are not likely to overfit the model to the data.

2.5.3. MODEL VARIANTS

In Figure 2.5, we compare the performance between the four variants of the proposed model in more detail, by changing the minimum amount of novel information n required to qualify sentences. For $n < 0.5$, the “no titles” variant obtains results that are very close to the “HTML titles” variant, indicating that most nuggets are also found in non-title sentences of the redundant news stream. The described approach does not use term weighting, except for the variant named “IDF weighted”. Our experiments show that using IDF for the estimation of similarity between news sentences hurts effectiveness. One observation is that relevant news sentences often contain numbers, however especially low numbers have relatively low IDF weights in most collections. Lastly, measuring the amount of previously unseen information using unigrams is less effective than using 2-word combinations.

The analysis of different variants shows that all variants outperform the existing systems for $n < 0.5$, indicating that using 3-NN clustering of sentences combined with the qualification of sentences against a relevance model over recently seen information does improve over current state-of-the-art approaches.

2.5.4. GROUNDTRUTH

According to the TREC definition, for the computation of Expected Gain, non-annotated sentences are ignored. For this study, only 5 of the 529 sentences in our main run had been annotated, which is clearly insufficient for a reliable estimation for both Expected Gain and Comprehensiveness, as can be seen in Table 2.2. Baruah et al. (2014) found that duplicate sentences in the KBA corpus have not been added to the official TREC ground truth. Therefore, for a system that returned a sentence that was annotated, results were different than for a system that returned a not annotated duplicate of that same sentence. They extended the official ground truth with duplicate sentences in the collec-

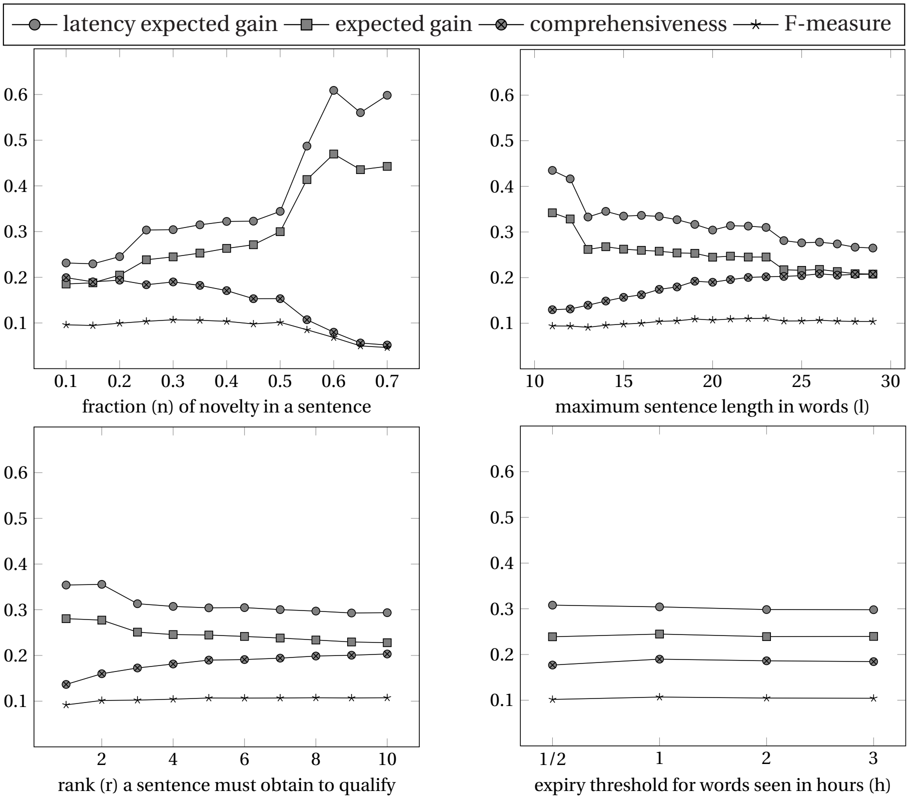


Figure 2.4: Impact of parameter values in the model performance, n=percentage of new word-pairs, not yet in the summary and co-occurring in the cluster, l=the maximum number of unique non stop words in a sentence, r=the minimum relevance rank amongst output sentences, h=number of (past) hours used to estimate the relevance model

tion, which we labeled the "Waterloo extended" set in Table 2.1. This extended ground truth set contains 38 of the sentences we returned. However, the results show an overestimation of Expected Gain, presumably because between systems there is more likely an overlap in relevant sentences than there is in non-relevant sentences. According to our observation, neither the official TREC ground truth nor the Waterloo extended set suffice for the evaluation of an external system; sentences missing in the existing ground truth would have to be annotated.

2.5.5. EXAMPLE OF CLUSTER IN ACTION

In Figure 2.6, we show a real example how a news article from the KBA corpus was processed for topic 4 "sikh temple shooting", from an article from which 2 sentences qualify for emission using the default ranking requirement $r = 5$. At 6:28pm a new article arrives, for which a node is added to the the title clustering. The nearest neighbors for the nodes are updated, and the new node forms a bi-directional loop with two of its nearest

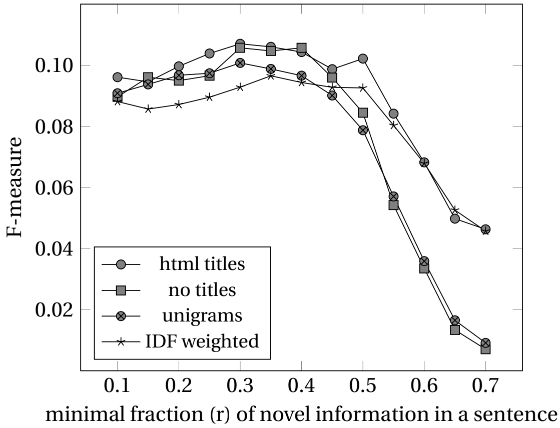
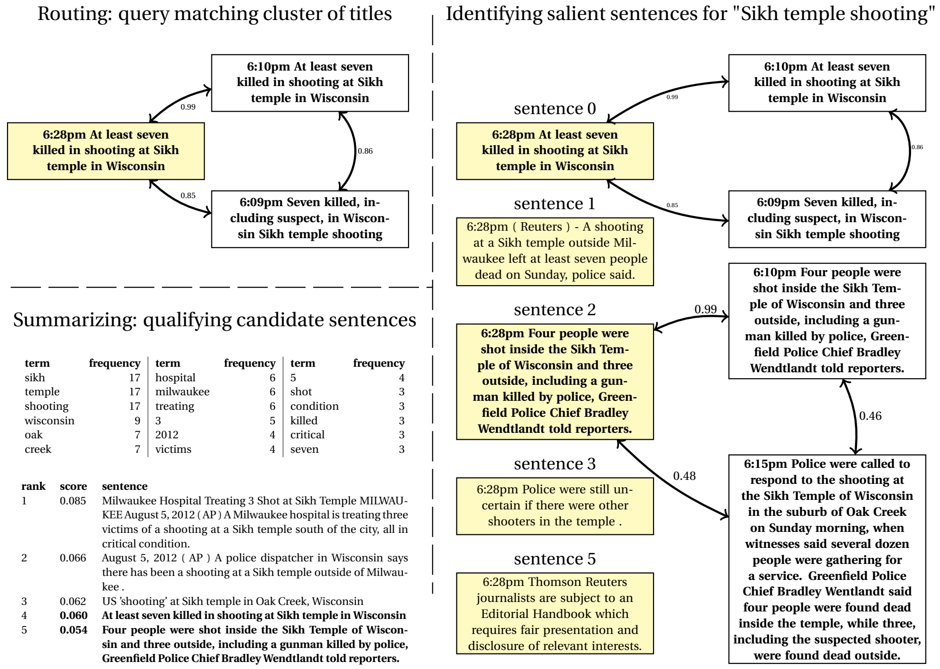


Figure 2.5: Comparison of the F-measure of model variants over different minimal amount of novelty in a sentence (n), a variant that is allowed to emit html titles, a variant that does not emit titles, a variant that estimates novelty using unigrams instead of word combinations, and a variant that uses IDF to estimate the cosine similarity for clustering sentences.

Table 2.2: Comparison of the performance of our HTML titles run, over the official TREC ground truth, the Waterloo extended set and a fully annotated set.

Ground truth set	Expected Gain	Latency DEG	Comprehension	F-Measure
TREC official	0.3224	0.4337	0.0149	0.014
Waterloo extended	0.2741	0.3640	0.0356	0.032
Fully annotated	0.2449	0.3019	0.1901	0.107

neighbors, thus a new cluster is formed. At least one of the cluster members contains all query terms in its title, therefore all articles in the ‘query matching cluster’ are routed to the sentence clustering graph for that query. To this graph, all sentences in the articles of the ‘query matching cluster’ are added, but only the article of the current document (6:28pm) are candidates to be added to the summary. Two candidate sentences become a member of a sentence cluster and therefore these are checked if they qualify. First, the relevance model is updated by removing ‘expired’ sentences and adding the core node sentences that are not a candidate sentence. Then the candidate sentences are ranked in a list with the sentences already in the summary using to the relevance model. In this example, both sentences satisfy the requirements for novelty, old information and rank in the top-5. The qualified sentences are added to the summary and the candidate sentences are added to the relevance model.


Routing: query matching cluster of titles
6:10pm At least seven killed in shooting at Sikh temple in Wisconsin
0.99
6:28pm At least seven killed in shooting at Sikh temple in Wisconsin
0.86
0.85
6:09pm Seven killed, including suspect, in Wisconsin Sikh temple shooting
Identifying salient sentences for "Sikh temple shooting"
sentence 0
6:28pm At least seven killed in shooting at Sikh temple in Wisconsin
0.99
6:10pm At least seven killed in shooting at Sikh temple in Wisconsin
0.86
0.85
sentence 1
6:28pm (Reuters) - A shooting at a Sikh temple outside Milwaukee left at least seven people dead on Sunday, police said.
6:09pm Seven killed, including suspect, in Wisconsin Sikh temple shooting
sentence 2
0.99
6:28pm Four people were shot inside the Sikh Temple of Wisconsin and three outside, including a gunman killed by police, Greenfield Police Chief Bradley Wendtlandt told reporters.
6:10pm Four people were shot inside the Sikh Temple of Wisconsin and three outside, including a gunman killed by police, Greenfield Police Chief Bradley Wendtlandt told reporters.
0.46
0.48
sentence 3
6:28pm Police were still uncertain if there were other shooters in the temple .
6:15pm Police were called to respond to the shooting at the Sikh Temple of Wisconsin in the suburb of Oak Creek on Sunday morning, when witnesses said several dozen people were gathering for a service. Greenfield Police Chief Bradley Wentlandt said four people were found dead inside the temple, while three, including the suspected shooter, were found dead outside.
sentence 5
6:28pm Thomson Reuters journalists are subject to an Editorial Handbook which requires fair presentation and disclosure of relevant interests.
Summarizing: qualifying candidate sentences

term	frequency	term	frequency	term	frequency
sikh	17	hospital	6	5	4
temple	17	milwaukee	6	shot	3
shooting	17	treating	6	condition	3
wisconsin	9	3	5	killed	3
oak	7	2012	4	critical	3
creek	7	victims	4	seven	3

rank	score	sentence
1	0.085	Milwaukee Hospital Treating 3 Shot at Sikh Temple MILWAUKEE August 5, 2012 (AP) A Milwaukee hospital is treating three victims of a shooting at a Sikh temple south of the city, all in critical condition.
2	0.066	August 5, 2012 (AP) A police dispatcher in Wisconsin says there has been a shooting at a Sikh temple outside of Milwaukee .
3	0.062	US 'shooting' at Sikh temple in Oak Creek, Wisconsin
4	**0.060**	**At least seven killed in shooting at Sikh temple in Wisconsin**
5	**0.054**	**Four people were shot inside the Sikh Temple of Wisconsin and three outside, including a gunman killed by police, Greenfield Police Chief Bradley Wendtlandt told reporters.**

Figure 2.6: A concrete example to illustrate data processing. *Routing.* A new article arrives at 6:28pm, and its title is added to a nearest neighbor graph of all existing titles. After assigning its nearest neighbors, it is part of a query matching (title) cluster, and therefore is routed to identify salient sentences. *Identification.* All sentences in the query matching cluster are added to the query's sentence clustering graph, the sentences from the current document being candidate sentences for the summarization. Two candidate sentences are clustered in sentence clusters that match the query and thus forwarded to the summarization of news for that query. *Summarize.* The core node sentences from the query matching sentences are added to the relevance model. The candidate sentences are ranked with the sentences that were already used in the summary. Both candidate sentences qualify because they are comprehensible (limited length and containing old information), contain a sufficient amount of novel information and rank in the top-5. The two qualified sentences are added to the summary and the candidate sentences are added to the relevance model.

2.6. CONCLUSION

In this study, we propose an approach for online temporal summarization of news related to ad-hoc information needs, expressed as a user query. In this approach, sentences are clustered based on cosine similarity, proximity in publication time and being supported by different news providers. The news extraction proceeds in three phases, first the titles of all incoming news articles are clustered, then we select the clusters in which the query terms appear and cluster the sentences contained in the clustered articles, and finally qualify a sentence as output when it contains sufficient novel information and is more relevant than the top sentences already in the summary. Our approach requires no a-priori model that separates news containing sentences from other content for an event type or in general, and can therefore be used to extract relevant news facts without knowledge about the type of the news event, and requires no manual intervention and contains a small number of parameters that can be tuned in straightforward fashion.

We evaluated the performance against the best systems using the 2013 TREC Temporal Summarization track test set. Our approach significantly improved results over the existing systems in F-measure and Latency-Discounted Expected Gain. Results indicate that news on average is reported before it was added to Wikipedia. Since in the crawled collection the publication time was estimated to be the crawl time, it is reasonable to expect further improvement in latency for a system that monitors news sites in real-time for new publications.

We explain the effectiveness of the approach by our focus on information that is support by several news providers and has a strong relatedness to the original query. However, as described, this approach is also likely to have limitations regarding the recall that can be obtained. Specifically, the requirement that a cluster contains a sentence that contains all words in the query makes the method more suitable to minimal queries than for elaborate queries or when the topic is likely to described using alternative words. An interesting direction for future work is to study how these constraints may be alleviated to improve recall.

3

ONLINE NEWS TRACKING FOR AD-HOC QUERIES

Following news about a specific event can be a difficult task as new information is often scattered across web pages. An up-to-date summary of the event would help to inform users and allow them to navigate to articles that are likely to contain relevant and novel details. We demonstrate an approach that is feasible for online tracking of news that is relevant to a user's ad-hoc query.

Keywords: *Information Filtering · Clustering · Multi-document summarization*

The contents of this chapter has been published as J.B.P. Vuurens, A.P. de Vries, R. Blanco and P. Mika, In Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, pages 1047–1048, ACM, 2015

3.1. Introduction

Internet users are replacing traditional media sources such as newspapers or television shows more frequently by online news. Although the Web offers a seemingly large and diverse set of information sources ranging from highly curated professional content to social media, in practice most sources base their stories on previously published works and add a much more limited set of new information. Therefore, users that seek additional information on a topic, often end up spending significant amount of effort rereading the same parts of a story before finding relevant and novel information. In such cases, an up-to-date summary of the event would help to inform users and allow them to navigate to articles that are likely to contain relevant and novel details. Online summarization is a crucial aspect of real-world products such as online live streams for natural disasters, product launches, financial or political events, breaking news notifications on mobile devices and topical daily news summaries like the Yahoo! news digest (`https://mobile.yahoo.com/newsdigest`).

In this demonstration, we suggest an alternative for tracking the news via Twitter hashtag subscriptions or Google Alerts. Compared to these existing approaches, the user is presented with a summary that contains the most important previously unseen facts along a timeline, topically related to a predefined ad-hoc query. The result is a timeline that is less redundant than Twitter and more insightful than the stream of headlines on Google Alerts. Instead of reporting social media utterances, we propose to give updates directly from the journalists' writings, by selecting relevant sentences with novel information from the news articles themselves.

The main contribution of this work is to demonstrate an approach that is feasible to tailor continuous news updates to ad-hoc queries. The technical contribution is to apply a multi-step three-nearest-neighbors clustering approach that can keep up with all the news arriving from hundreds of RSS feeds. For each of those feeds, we fetch the full articles from the source, and process their contents to identify the most novel and relevant sentences. The resulting system participates in the TREC temporal summarization task, but the demonstration gives more interpretable and comparable results than the TREC evaluation measures indicate. Also, users of the demo can track their own information needs.

3.2. Existing work

The system used can be viewed as a hybrid combination of techniques for query based online news tracking and summarization, adapted from Allan et al. (1998, 2001). A side effect of broadcast news is that stories with near identical publication times are more likely to discuss related events (Allan et al., 1998), which is what we use in this study to cluster news. Our approach differs by a stronger emphasis on novelty of information emitted (e.g. Gabrilovich et al., 2004). Hereto, we estimate the amount of previously unseen information to use only sentences that are likely to contain novel information.

3.3. Tracking Ad-hoc Requests

For online news tracking, we propose to use articles that are published on online news sites. The title of news articles can be viewed as a short summarization of its content, and therefore used to determine if articles are likely to describe the same topic. Given the relative low-memory requirements of news headlines, this allows for fast in memory clustering without the need to partition the data. The publication of most news articles can be monitored using RSS feeds, which allows fast access to the articles' title and publication time.

In this demo, we summarize a stream of online news articles in a three-step process: *cluster titles*, *cluster sentences* and *qualify sentences*. In the first step, titles of newly published articles are continuously downloaded from RSS feeds. These titles are connected to their nearest neighbors based on similarity measure that considers their title and publication time. Salient sentences are found in sentence clusters of the nearest neighbor graph, when sentences from at least three different sources are most similar, indicating news facts that are more interesting rather than (opinion) information that is not supported. In step 2, per ad-hoc query a graph of sentences is created and maintained. The output of step 1 is monitored, and if a *query matching cluster of titles* is formed or modified, i.e. that contains a title that includes all query terms, then all sentences of the news articles in that cluster are added to the sentence graph of the ad-hoc query. Finally, in step 3, the *qualifying* sentences in the arriving news article are emitted to the user. For qualification, a sentence must (a) be among a cluster of at least three sentences from different news domains, (b) be ranked in the top-K of emitted sentences using a relevance model over the information seen in the last hour, and (c) add information previously not shown to the user.

3.4. FEASIBILITY

In this demo, we show that online news tracking can be done with reasonable latency in commodity machines. Typically, the monitoring of RSS feeds and maintaining a nearest neighbor graph over news headlines takes less than 10% of the capacity of a standard computer. The clustering and qualifying of sentences is processed independently for different ad-hoc queries and therefore can be processed in parallel and scaled up in production systems. Additionally, efficiency can be improved when the news timelines for known entities (e.g. Wikipedia) are cached in advance, allowing steps 2 and 3 for entity related queries to be simplified to a filtering task over a cached timeline.

3.5. DEMONSTRATION

We provide three ways to participate in the demo. At the stand, we will show a summary of topics that are trending at that time. For these summaries, the information is processed as if online and therefore represent what the user would have received when they subscribed to the query at the first update. Additionally, participants can experience receiving new updates on the topics they wish to track, by subscribing to a live generated RSS feed for current trends. Finally, we provide limited opportunity to enter ad-hoc queries, depending on the resources needed to downloading the articles for new topics.

In Table 3.1, we show an example of a time line constructed for the query "Copenhagen", after a terrorist attack on Februari 14th 2015. The first mention on Twitter of the hashtag #CopenhagenShooting was at 17:11 (`http://ctrlq.org/first/`), and the first information was added to Wikipedia at 17:06. A larger static example of the demo

Table 3.1: Timeline constructed for the query "Copenhagen" from Feb 14 2015 16:19.

Time	Sentence
2015-02-14 16:19:58	Copenhagen - Shots were fired on Saturday near a meeting in the Danish capital of Copenhagen attended by controversial Swedish artist Lars Vilks, Sweden's TT news agency reported.
2015-02-14 16:49:22	COPENHAGEN, Denmark - At least one gunman opened fire Saturday on a Copenhagen cafe, killing one man in what authorities called a likely terror attack during a free speech event organized by an artist who had caricatured the Prophet Muhammad.
2015-02-14 17:34:26	COPENHAGEN, Denmark (AP) – A gunman fired on a cafe in Copenhagen as it hosted a free speech event Saturday, killing one man, Danish police said.
2015-02-14 18:51:04	After searching for the gunman for hours, police reported another shooting near a synagogue in downtown Copenhagen after midnight.
2015-02-14 19:29:41	One person was shot in the head and two police were wounded in an attack on the synagogue in central Copenhagen, Danish police said, adding that it was too early to say whether the incident was connected to an earlier one at an arts cafe.
2015-02-15 01:56:20	French President Francois Hollande called the Copenhagen shooting “deplorable” and said Thorning-Schmidt would have the “full solidarity of France in this trial.”
2015-02-15 03:52:40	Denmark was on high alert and a massive manhunt was under way on Sunday after a man sprayed bullets at a Copenhagen cafe hosting a debate on freedom of speech and blasphemy, killing one person and wounding three police officers.

results can be viewed online, at `http://newstracker.github.io/`.

4

Hierarchy construction for news summarizations

Following online news about a specific event can be a difficult task as new information is often scattered across web pages. In such cases, an up-to-date summary of the event would help to inform users and allow them to navigate to articles that are likely to contain relevant and novel details. Several approaches exist to compose a summary of salient sentences that are extracted from an online news stream for a given topic. Summaries often consist of multiple news stories, that when entwined may make it harder to read. We propose a general approach to convert non-hierarchical temporal summarizations into a hierarchical structure, that can be used to further compress the summary to provide more overview, that allows the user to navigate to specific subtopics of interest, and can be used to provide feedback to improve results. This approach reorganizes the sentences in a summary using a clustering approach to capture the sentences per news story in a hierarchy.

***Keywords**: Hierarchical Clustering · Multi-document summarization*

The contents of this chapter has been published as J.B.P. Vuurens, A.P. de Vries, R. Blanco and P. Mika, Hierarchy Construction for News Summarizations, In Proceedings of the SIGIR Workshop on Temporal, Social and Spatially-aware Information Access, 2015

4.1. Introduction

Internet users are turning more frequently to online news as a replacement for traditional media sources such as newspapers or television shows. Still, discovering news events online and following them as they develop can be a difficult task. Although the Web offers a seemingly large and diverse set of information sources ranging from highly curated professional content to social media, in practice most sources base their stories on previously published works and add a much more limited set of new information. Thus users often end up spending significant amount of effort re-reading the same parts of a story before finding relevant and novel information. Online summarization is a crucial aspect of real-world products such as online live streams for natural disasters, product launches, financial or political events, breaking news notifications on mobile devices and topical daily news summaries like Yahoo News Digest (`https://mobile.yahoo.com/newsdigest`).

Allan et al. (2001) formalize the temporal summarization problem as follows. A news topic is made up of a set of events and is discussed in a sequence of news stories. Most sentences of the news stories discuss one or more of the events in the topic. To construct a summary, significant updates are identified for the topics being tracked, relieving the users from having to sift through long lists of similar articles arriving from different news sources, and minimize the time and disruptions to users who wish to follow evolving news stories, similar to the proposal by Gabrilovich et al. (2004). For this purpose, several methods have been proposed (e.g. Gabrilovich et al., 2004; Radev et al., 2005; Tran et al., 2015; Vuurens et al., 2015a). This research focuses on news topics that are made up of multiple news stories, which are not necessarily discussed in sequence but can also overlap, e.g. for the topic Apple a news story about the iPhone patent case against Samsung can overlap with a news story about the new Apple Watch. This aspect is overlooked in the evaluation of other work for temporal summarizations, which is often focused on recall and precision-like metrics that do not consider the context in which a sentence is reported. Consider the following example:

> Mexico City Mayor Miguel Angel Mancera said many evacuations were reported in the capital but officials received no reports of damage or injuries.
>
> MEXICO CITY - A moderate 5.3-magnitude earthquake shook central Mexico on Friday, causing buildings to sway in the capital and sending hundreds of people into the streets.
>
> According to Pemex's official Twitter account, the platform, called Abkatun Permanente, caught fire early Wednesday in Campeche Sound in the Gulf of Mexico.

The first sentence refers to an event, and therefore is easier to understand when accompanied by a context that informs what this event is. If the first sentence is accompanied (or even better preceded) by the second sentence the information of the first sentence becomes clear. However, if news stories appear entwined and there are several possibilities, this can be more confusing and harder or even impossible to understand.

In this example this occurs when Sentence 3 appears close to sentence 1, since evacuations, damage and injuries could also be related to a fire on an oil platform. Therefore, a flat temporal ordering of extracted news sentences may be less efficient to read when a news stream contains entwined news stories, opposed to when there is only a single news story.

In this work, we aim to improve the comprehensiveness of a summary that contains multiple news stories by capturing each of the underlying news stories in a cluster of a constructed hierarchy. We experimented on query based timelines of sentences that were extracted from news articles. This approach may apply to a broader domain of temporal summarizations, which we leave to study in future work. The hierarchy is constructed by first separating the news articles into pools that are unlikely to discuss the same news story, and then per pool use an divisive clustering approach to further separate sentences that are not likely to refer to the same news. The produced hierarchy provides the user with a more concise overview over a more diverse set of news stories, allowing to drill down the hierarchy to view specific news stories. To evaluate our approach we report the F-Score obtained on a set of news summaries for which we annotated the news stories.

The remainder of this paper is structured as follows: Section 2 discusses related work, in Section 3 we describe our approach, Section 4 discusses the experiment setup, in Section 5 we report the results obtained and finally in Section 6 we present the conclusion.

4.2. Related Work

A common method for the temporal summarization over multiple texts is to extract their salient sentences, i.e. the sentences that are most *useful* and *novel*. In previous work, we proposed to select sentences from a stream of news articles based on their salience estimated by three factors: relatedness in the 3 nearest neighbor graph, the presence of information the user has not seen, and, a top rank among sentences in the summary when scored using the information seen over the last hour (Vuurens et al., 2015a). Similar to other news summarizers (e.g. Gabrilovich et al., 2004; Radev et al., 2005; Tran et al., 2015), the result is a chronologically ordered news summary.

The construction of a hierarchy for such a summary can be seen as a clustering approach over its contents. For instance, Quinlan (1986) summarizes an approach to synthesizing decision trees from information that is noisy and/or incomplete. For each attribute he proposes to choose the attribute with the highest Information Gain, i.e. that results in the lowest entropy when used to divide the data into subsets. Cheng et al. (1999) propose to use an entropy-based method for clustering, motivated by the fact that a subspace containing clusters typically has a lower entropy than a subspace without clusters. Liu et al. (2003) compare the cluster quality of several feature selection methods for K-means clustering and found that in an unsupervised experiment that Information Gain is one of the best feature selection methods, especially when only a small amount of features is selected. After analyzing the words that were selected by each method they concluded that Information Gain is more likely to select discriminative terms than other methods.

Hierarchic-clustering methods result in tree-like classifications in which small clusters of objects (i.e. documents) that are found to be strongly similar to each other are nested within larger clusters that contain less similar objects. Hierarchical-clustering methods are divided into two broad categories, agglomerative and divisive. Divisive methods normally result in monothetic classifications, where documents in a given cluster must contain certain terms in order to gain membership. On the other hand, in polythetic clustering methods no specific terms are required for membership in a cluster, and such structures are usually the result of agglomerative methods. For information retrieval, polythetic clusterings are preferred (Tombros, 2002). In this research, we propose a polythetic clustering method based on normalized Information Gain. To the best of our knowledge Information Gain has not been used in IR research before to cluster high-dimensional data in a noisy collection.

4.3. Design

We describe how to build a hierarchy for a summary of sentences that were selected from a collection or stream of news articles. We propose the construction as a two step clustering process: (1) separate the news articles that were used for the summary into *document pools* that are unlikely to discuss the same news, and (2) per pool, form clusters based on the most different sentences.

The objective for step (1) is to separate the news articles into pools so that articles that are likely to discuss the same news story are pooled together, while creating separate pools for articles that do not. For example, news articles that contain the word “apple”, may partly be related to the computer company and partly to fruit which should ideally be assigned to different pools. At the sentence level there is often insufficient information to make a reliable decision, therefore we use the articles from which the sentences in the summary were selected. The pooling is based on dissimilarity (or impurity) estimated by a normalized version of the *Information Gain*. Information Gain has been successfully used when considering a fixed number of non-sparse dimensions of a data set (Cheng et al., 1999; Quinlan, 1986), however, in text collections the Information Gain is not comparable between subsets that use a different number of features. We introduce *normalized Information Gain* to address this problem, a measure that returns 0 for identical subsets and 1 for disjoint subsets. Formally, in Equation 4.1 H is the entropy over the words w in a bag of words s, given the size of the content c and $f_{w,s}$ is the frequency of word w in s. In Equation 4.2, the Information Gain IG is defined for separating a group of content $s+t$ into two separate bags of words s and t, with $|s|$, $|t|$ and $|s|+|t|$ as the number of words contained. In Equation 4.3, IG_{max} is the maximum Information Gain that would be obtained given the word distribution of the subsets t and s if these are completely disjoint, and in Equation 4.4 IG is divided by IG_{max} to normalize IG_{norm} to a value in $[0,1]$. The normalized Information Gain can be computed between sentences and articles, and also between clusters of sentences and articles by considering these to be the concatenation of the contained elements.

$$H(s,c) = -\sum_{w \in s} \frac{f_{w,s}}{c} log_2 \frac{f_{w,s}}{c} \tag{4.1}$$

$$\begin{aligned} IG(s,t) =& H(s+t, |s|+|t|) \\ & - \frac{|s|}{|s|+|t|} \cdot H(s,|s|) - \frac{|t|}{|s|+|t|} \cdot H(t,|t|) \end{aligned} \tag{4.2}$$

$$\begin{aligned} IG_{max}(s,t) =& H(s,|s|+|t|) + H(t,|s|+|t|) \\ & - \frac{|s|}{|s|+|t|} \cdot H(s,|s|) - \frac{|t|}{|s|+|t|} \cdot H(t,|t|) \end{aligned} \tag{4.3}$$

$$IG_{norm}(s,t) = \frac{IG(s,t)}{IG_{max}(s,t)} \tag{4.4}$$

We build the hierarchy as follows. In step (1), the news articles from which sentences were used in the summary are processed in order of publication time, comparing the IG_{norm} of a new article to the existing pools of articles, adding it to the pool with which it has the lowest IG_{norm} if this is below a threshold ω_d, or otherwise creating a new pool. Pools are merged when the IG_{norm} between them becomes lower than this threshold.

In step (2), we consider the news articles' sentences that were selected for the summary in a single pool. We first create a set S of most different *sentences* with between them an IG_{norm} that exceeds the threshold ω_s. For this, we sort all pairs of sentences by their IG_{norm}, and greedily add the next pair that has the next highest IG_{norm} to S as long as the IG_{norm} between all sentences in S exceed ω_s, or otherwise skipping over to the next pair in the sorted list. The sentences in S form the initial clusters, and we iteratively add an unassigned sentence from the pool that has the lowest IG_{norm} with any of the clusters to that cluster. When all sentences have been assigned, clusters are merged when the IG_{norm} between them is below the threshold ω_s. If a pool contains several clusters, the pool is considered to be parent node to the sentence clusters, otherwise a pool is considered to be a single non-hierarchical cluster.

4.4. EXPERIMENT

To evaluate whether the constructed hierarchy separates the news stories that are contained in the temporal summarization for a query, we use F-Score as proposed by Larsen and Aone (1999). For this metric, the cluster hierarchy is treated as an output from an automatic multi-level routing system, in which for each news story a corresponding cluster will form automatically somewhere in the hierarchy. A parent cluster contains the information of its sub-clusters, therefore different hierarchy levels are tried to find the level that is the best match for each news story. Formally, in Equation 4.5, $f(t)$ is the total number of sentences that match news story t over all clusters C in the hierarchy, in Equation 4.6 the precision for t in C is defined as the number of sentences in C that match news story t divided by the number of sentences in the cluster $|C|$, and in Equation 4.7 the recall is defined as the number of sentences in C that match the news story t divided by the total number of sentences that match that news story in the hierarchy. Then in Equation 4.8, for the computation of F, the clusters are considered to include the information of its subclusters, and the F for a news story t is the maximum F-measure over all clusters. In Equation 4.9, the $F-Score$ is the weighted average of F-measures over all annotated news stories.

$$f(t) = \sum_{C} |\{s \in C | s.topic = t\}| \tag{4.5}$$

$$precision(C,t) = \frac{|\{s \in C | s.topic = t\}|}{|C|} \tag{4.6}$$

$$recall(C,t) = \frac{|\{s \in C | s.topic = t\}|}{f(t)} \tag{4.7}$$

$$F(t) = \max_{C} \frac{2 \cdot precision(C,t) \cdot recall(C,t)}{precision(C,t) + recall(C,t)} \tag{4.8}$$

$$F-Score() = \frac{\sum_t f(t) \cdot F(t)}{\sum_t f(t)} \tag{4.9}$$

$$\tag{4.10}$$

The data set for this evaluation consists of temporal summaries for the queries "apple", "mexico", "cyclone" and "volcano", that were tracked by NewsTracker (Vuurens et al., 2015a) over the period between March 1st 2015 and June 1st 2015 using the articles published by 70 online news providers (e.g. CNN, NY Times, BBC News). For the original summaries, we identified the news stories contained and assigned every sentence to one news story. We identified both entity related news stories such as a named cyclone, a volcano, an apple product, as well as event related news stories such as "patent lawsuit apple samsung", "Jalisco Cartel boss arrested". Some news can both be considered a separate news story or part of a bigger news story, as a rule of thumb we annotated the level at which the other sentences provide a useful context, e.g. a violent response to the arrest of the Jalisco Cartel would not be a separate new story reasoning that the information about the arrest provides a context that makes the news about the subsequent violence easier to read. In Table 4.1 we report the number of sentences and news story per query. All news stories were annotated, including for instance the story “an apple a day does not keep the doctor away” for the query “apple”. Although we were not interested in this particular story, we focus this evaluation on clustering quality and therefore annotated these news stories as well, and in practice, correctly clustered irrelevant information may hurt the comprehensiveness of a summary less than scattered irrelevant information. Sentences that are not related to a news story are counted in the cluster size and therefore discounts the precision for the cluster they appear in. The news summarizations used for input and the annotated ground truth can be found at `http://hns-dataset.github.io/`. Note that we do not evaluate the effectiveness of the summary used as input.

Although for the task we defined there is no state-of-the-art baseline available to compare with, we add a comparison to two simple baselines as reference of the obtained results. The “Single Linkage” method clusters sentences based on nearest neighbor clustering where each sentence is assigned the sentence with the highest cosine similarity between a TF-IDF representation of their contents. The “Multiple Linkage” links all sentences that have a cosine similarity between them that exceeds a threshold. For the latter,

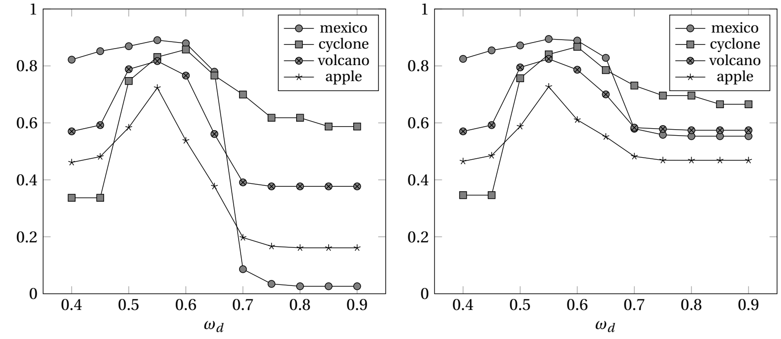
Table 4.1: Queries used for evaluation with the number of sentences in the summary and the number of news stories identified.

Query	Sentences	News stories
Cyclone	87	8
Volcano	90	8
Apple	373	60
Mexico	552	145

we chose the threshold that obtained the highest F-Score per query. For both methods, the connected subgraphs represent the clusters that were formed.

4.5. RESULTS

We first inspect the effectiveness of document pooling (step 1) in those runs in which we did not use step 2. In Figure 4.1a, we compare the F-Score per query of each resulting hierarchy when varying the threshold that is to pool documents. Naturally, extreme values for ω_d do not score well, since setting it too high creates a single pool only, without hierarchy, while low values of ω_d establish a separate pool per document. We observe a query dependent “sweet zone”, which is possibly related to the coherence between separate news stories of a query; which in our observation was lower for “mexico” than for “apple”. In this experiments, a setting for ω_d of 0.55 is (close to) optimal for every query.



(a) when not using step (2).

(b) when using step 2 with a fixed setting $\omega_s = 0.75$.

Figure 4.1: Comparison of the F-Score per query when varying ω_d.

Next, we include step (2) and plot the average F-Score over all four queries for a sweep of ω_d and ω_s (Figure 4.2). The results indicate that when the pooling parameter ω_d has an optimal setting, the effect of the grouping parameter ω_s does not further improve clustering quality. However, when ω_s is set too high and unrelated news articles are pooled together, the creation of sub-clusters within its pool can salvage some of the coherence that is lost. In this experiment, setting ω_d to a value of 0.75 - 0.80 maximizes results for cases with too coarse pooling granularity. By comparing Figure 4.1a with Fig-

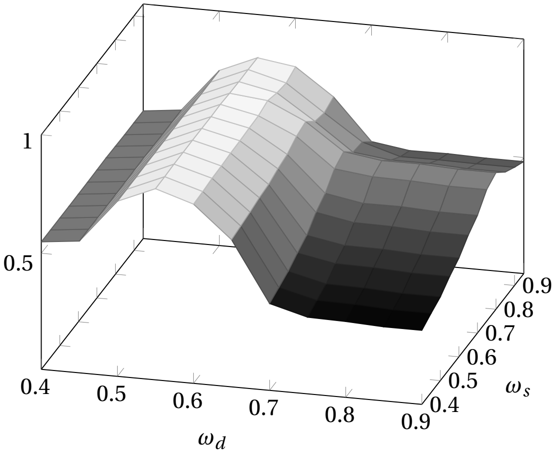


Figure 4.2: Comparison of the average F-Score over all queries when varying ω_d and ω_s.

Table 4.2: The F-Score obtained when optimal parameter settings are found for the proposed approach and the Single Linkage and Multiple Linkage baselines.

Query	ω_d	ω_s	F-Score	Single Linkage	Multiple Linkage
Cyclone	0.60	0.75	0.867	0.600	0.644
Volcano	0.55	0.60	0.835	0.425	0.410
Apple	0.55	0.90	0.729	0.458	0.385
Mexico	0.55	0.75	0.895	0.562	0.563

ure 4.1b we see that step (2) is only effective when ω_s is overestimated, and when used with a fixed threshold $\omega_s = 0.75$ this improves the clustering quality.

In Table 4.2, we list the F-Score that is obtained when the optimal parameter settings are used for each query. For reference, we include the F-Score obtained when using Single Linkage and Multi Linkage. The results indicate a potential of normalized Information Gain to cluster textual summaries. However, these results are only realistic if in practice near optimal settings for the parameters can be found. An interesting direction for future work is to test the stability of these parameters over a larger collection, or when they appear not stable to predict ideal settings per query, for instance based on observed differences between news articles.

4.6. CONCLUSION

In this study we propose a novel approach to hierarchically cluster news stories that are part of a news summary based on separation of unrelated news as measured by their normalized Information Gain. We evaluate this approach by assessing the cluster quality produced by our method which was measured by the F-Score over the constructed hierarchy. We compared our approach using a ground truth of news stories that were labeled to the sentences in the summary. We conclude that this approach has the potential to accurately capture news stories in a hierarchy. An interesting direction for future work is to personalize a news summary using the clusters obtained.

5

First story detection using multiple nearest neighbors

First Story Detection (FSD) systems aim to identify those news articles that discuss an event that was not reported before. Recent work on FSD has focussed almost exclusively on efficiently detecting documents that are dissimilar from their nearest neighbor. We propose a novel FSD approach that is more effective, by adapting a recently proposed method for news summarization based on 3-nearest neighbor clustering. We show that this approach is more effective than a baseline that uses dissimilarity of an individual document from its nearest neighbor.

Keywords: *Document Filtering*

The contents of this chapter has been published as J.B.P. Vuurens, A.P. de Vries, In Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval, pages 845–848, ACM, 2016

5.1. INTRODUCTION

Internet users are turning more frequently to online news as a replacement for traditional media sources such as newspapers or television. For the user, the news stream is a source to both track topics of interest and to become informed about important new events the user was not yet aware of. Automated detection of new events can save to user a great deal of time, for instance by notifying users about new events, which is especially interesting to users and organizations for whom the information is time-critical and who need to act on that information.

First Story Detection (FSD) systems aim to identify those news articles that discuss an event that was not reported before in earlier stories, without knowledge of what events will happen in the news (Allan et al., 1998). Recently, FSD has been suggested as a useful tool to monitor the Twitter feed by Petrović et al. (2010), and while previous work has addressed the efficiency that is required for this purpose, there has been little work on improving the effectiveness in over a decade (Petrović et al., 2010, 2012).

In this study, we propose a novel approach that is more effective than the widely used function proposed by Allan et al. (2000) that declares a story new if it is dissimilar to its nearest neighbor.

5.2. RELATED WORK

The task of detecting events can be automated using information about the events published online. For this purpose, the Topic Detection and Tracking (TDT) program was initiated to discuss applications and techniques to organize broadcast news stories by the real world events that they discuss in real-time. News stories are gathered from several sources in parallel to create a single stream of constantly arriving news. The problem of first story detection is to identify the stories in a stream of news that contain discussion of a new topic, i.e. whose event has not been previously reported (Papka and Allan, 2002).

FSD has been recognized as the most difficult task in the research area of TDT (Yang et al., 2002). In early work, Allan et al. (2000) detect first stories as news articles whose cosine similarity over tf-idf vectors to its nearest neighbor is less than a threshold, an effective approach that outperforms complex language model approaches in most cases. This baseline is still used for FSD in recent work, in which more focus is put on efficiency than to improve effectiveness (Karkali et al., 2013; McCreadie et al., 2013; Osborne et al., 2012).

Papka and Allan (2002) and Allan et al. (1998), argue that a side-effect of the timely nature of broadcast news is that stories closer together on the news stream are more likely to discuss related topics than stories farther apart on the stream. When a significant new event occurs, there are usually several stories per day discussing it; over time, coverage of old events is displaced by more recent events. They use temporal proximity as a distinguishing feature to incorporate the salient properties of broadcast news.

In recent work, Vuurens et al. (2015c) proposed a novel 3-nearest neighbor clustering (3NN) approach to retrieve sentences from news articles that contain novel and useful news facts. In this approach every text is linked to its three nearest neighbors that must be from a different domain. The so-called '2-degenerate cores' constructed by the algorithm correspond to highly similar texts from different sources. Their existence indicates the importance or salience of the information contained. Temporal proximity is incorporated in the model by weighting the time between news articles in the similarity function used. In Vuurens et al. (2015b) normalized information gain is shown to be more effective than cosine similarity for the task of clustering news articles that are topically related.

5.3. METHOD

In this work, we adapt the 3NN clustering approach to First Story Detection, by clustering news articles rather than sentences, and using a similarity function based on normalized information gain to promote the clustering of news articles that are likely to be topically related.

5.3.1. SINGLE LINKAGE

We compare our efforts to the approach described by Allan et al. (2000), which is considered a state-of-the-art approach in recent studies on First Story Detection (e.g. McCreadie et al., 2013; Petrović et al., 2010). In this approach, documents are represented as tf-idf weighted vectors, and the novelty of a document d is estimated by the cosine similarity to its nearest neighbor n in the collection C (Allan et al., 2000):

$$novelty(d) = 1 - \max_{n \in C} cos(d, n) \tag{5.1}$$

Then, a news article is marked as a first story when its novelty is below a threshold $\alpha \in [0, 1]$.

5.3.2. 3NN FIRST STORY DETECTION

In this study, we propose a novel approach that is based on 3-nearest neighbor clustering (3NN), using the existing open source implementation (Vuurens et al., 2015c). In 3NN clustering, every node is assigned to its three nearest neighbors, not allowing links between nodes from the same news domain, and based on temporal proximity between publication dates which allows the clustering to be continuously updated in near real-time. 2-generate cluster cores are formed when three nodes each link to the other two as a one of its 3 nearest neighbors. These clusters contain information that is locally most central and therefore likely to be salient information (Vuurens et al., 2015c). The key idea for First Story Detection, is that acting on formed 3NN clusters rather than individual documents is less likely to return false positives. However, instead of truly detecting the first story as was the objective in the TDT program, here we aim to improve detection performance at the expense of slightly delayed detection. It may also be that the story detected as the first of a new event is more central to the information, and therefore more suitable as a seed to start tracking a topic, however, this hypothesis is outside the scope of this study and left for future work.

In Vuurens et al. (2015b), news sentences were fitted into a hierarchy that distinguishes between different events and topics by forming clusters of topically related the news articles, for which normalized information gain was shown to be more effective than cosine similarity. Therefore, to promote 3NN clusters to be formed around topically related news articles we use a similarity function based on normalized information gain. In Equation 5.2, the normalized information gain between two documents d and d' results in a score of 0 between identical documents and a score of 1 between disjoint documents, by dividing the information gain IG between the documents by an upper bound of the information gain IG_{max} that would be obtained if these documents have the same internal distributions over terms but are completely disjoint. For the remainder of this paper we use IG_{sim} as defined in Equation 5.3 as a similarity function between two documents d, d' based on IG_{norm}.

$$IG_{norm}(d,d') = \frac{IG(d,d')}{IG_{max}(d,d')} \tag{5.2}$$

$$IG_{sim}(d,d') = 1 - IG_{norm}(d,d') \tag{5.3}$$

From the obtained 3NN clustering, the newly formed 2-degenerate cores are inspected for first stories. Similar to the Single Linkage baseline, first stories are detected when a newly formed cluster core is dissimilar from news articles seen recently. In 3NN every news article is linked to its three nearest neighbors, therefore the members of a newly formed 2-degenerate core that contains a first story each have two links to the other core members and the third link links to a dissimilar news article. The most similar non-core news article that a core member links to, is then used to estimate the novelty of that cluster core. Formally, in Equation 5.4 a cluster core A is declared novel when the similarity between a news article $d \in A$ and a news article n in the remainder of the collection C is below a threshold $\phi_{novelty}$.

$$novelty(A) = \max_{d \in A, n \in C-A} IG_{sim}(d,n) < \phi_{novelty} \tag{5.4}$$

Lastly, we add a threshold to filter out newly formed clusters that are less likely to be topically related to each other. Vuurens et al. (2015b) show that news articles that have a high normalized information gain are rarely topically related. Following their findings, we filter out clusters that fail the coherence criterium in Equation 5.5, that enforces that the similarity between all nodes d, d' that are members of the same 2-degenerate core A exceeds a threshold $\phi_{coherence}$. Different settings for this threshold are tried to examine the sensitivity and impact on effectiveness.

$$coherence(A) = \min_{d \in A, d' \in A-\{d\}} IGsim(d,d') > \phi coherence \tag{5.5}$$

5.3.3. Test set

For the evaluation, we use the TREC Temporal Summarization Track test sets of 2013 and 2014. The corpus for these test sets is the 2013 TREC KBA Streaming corpus, which contains approximately 150M news articles that are processed in a strict online setting. Table 5.1 shows the topics from the test sets, which are all types of a crisis that received continuous updates in the media over time. Arguably, the news regarding a single topic could be considered to be all part of the same story, or in some cases be regarded as separate stories within a topic. Here we regard all news articles that are matched to the same topic as part of one news story, for which ideally only the first article should be returned. TREC assessors annotated the sentences that TREC participants retrieved as relevant if they contain a news fact relevant to the topic.

The basis for the evaluation of the FSD systems is a list per topic of all documents that contain relevant news facts according to the TREC ground truth or the online published extended lists that contain duplicate sentences found in the collection. For the combined 23 topics, there are 65,358 documents that were annotated as containing relevant information. For this task, a returned news article is considered as a first for a topic when it is the first relevant article returned by the system, and a false alarm when another relevant article for the same topic was returned earlier. News articles that are not marked as relevant to the topic are ignored in the evaluation.

Table 5.1: Topics for the 2013 and 2014 TREC TS Track

Topic	Title
1	2012 Buenos Aires Rail Disaster
2	2012 Pakistan garment factory fires
3	2012 Aurora shooting
4	Wisconsin Sikh temple shooting
5	Hurricane Isaac (2012)
6	Hurricane Sandy
8	Typhoon Bopha
9	2012 Guatemala earthquake
10	2012 Tel Aviv bus bombing
12	Early 2012 European cold wave
13	2013 Eastern Australia floods
14	Boston Marathon bombings
15	Port Said Stadium riot
16	2012 Afghanistan Quran burning protests
17	In Amenas hostage crisis
18	2011-13 Russian protests
19	2012 Romanian protests
20	2012-13 Egyptian protests
21	Chelyabinsk meteor
22	2013 Bulgarian protests against the Borisov cabinet
23	2013 Shahbag protests
24	February 2013 nor'easter
25	Christopher Dorner shootings and manhunt

5.3.4. EXPERIMENT SETUP AND EVALUATION METRICS

The effectiveness of First Story Detection systems is measured by the miss rate, false alarm rate, recall and precision, which we explain using the contingencies in Table 5.2. For any topic, we only consider articles that are annotated as relevant for the topic, thus if T is the number of documents annotated as relevant for the topic, then $TP+FN+FP+TN=T$. Since there can only be one first story per topic per system, $TP+FN=1$ and $FP+TN=T-1$. A miss occurs when the system fails to detect a new event, i.e. *miss rate* = $\frac{FN}{TP+FN}$. A false alarm occurs when the system emits a news article when a first story was already emitted for that topic, i.e. *false alarm rate* = $\frac{FP}{FP+TN}$. Recall is the fraction of topics for which a first story was detected *recall* = $\frac{TP}{TP+FN}$, and precision is the fraction of retrieved news articles that is a fist story *precision* = $\frac{TP}{TP+FP}$, which here only considers the news articles that are relevant to the topic.

Table 5.2: Contingency table for evaluation metrics

	Retrieved	Not retrieved
First story	TP	FN
Not first story	FP	TN

5.4. Results

In this Section, we compare the effectiveness of first story detection using Single Linkage (SL) to FSD using 3NN.

5.4.1. Effectiveness

In Figure 5.1, a DET curve shows the relationship between miss rate and false alarm rates. Overall, the 3NN runs perform better than SL, regardless of the setting used for $\phi_{coherence}$. In Figure 5.2, we show a tradeoff between recall and precision, which further supports that 3NN is consistently more effective than Single Linkage. Table 5.3 gives the precision and false alarm rate when the novelty thresholds for both systems are set to the highest precision that can be obtained at recall = 1. When $\alpha = 0.48$ and $\phi_{novelty} = 0.6$ are set to allow for the lowest false alarm rate at a missed rate of 0 (i.e. recall=1), precision is respectively 0.0149 for SL and 0.0618 for 3NN, meaning that SL more redundantly retrieves 4 times more news articles for the same event.

Table 5.3: Optimal effectiveness at recall=1.

	precision	false alarm rate
Single Linkage $\alpha = 0.48$	0.0149	0.0195
3NN $\phi_{novelty} = 0.60$	0.0618	0.0053

5.4.2. Timeliness

In Figure 5.3, the y-axis shows the aggregated number of relevant news articles per hour over time on the x-axis. In this Figure, we can visually compare the moment a first story was detected against the volume of published news articles. We can see that the systems occasionally missed early detection, e.g. 3NN for topic 3, and Single Linkage for topic 9. On topic 12, detection may be late for 3NN, but there is a difficult tradeoff between early detection and a lower false alarm rate.

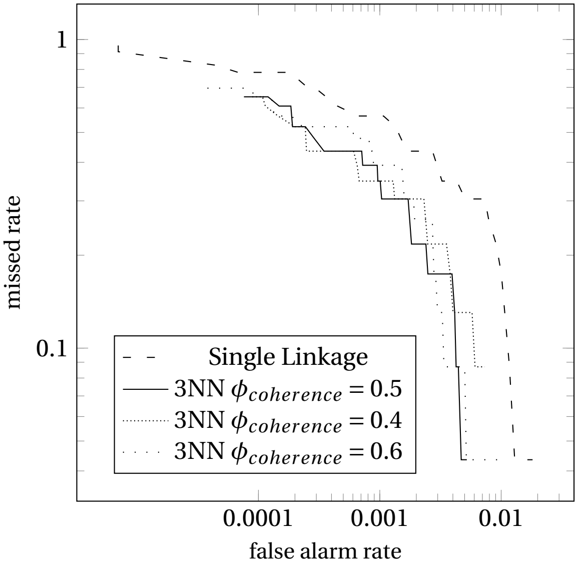


Figure 5.1: Detection Error Tradeoff curve, closer to the origin is better.

Some topics are related to an incident that is followed by a quick burst (e.g. topic 1), while other topics initially have a phase of little media attention and have intervals of increased interest later in time (e.g. topic 16). An interesting case is topic 18, which concerns the demonstrations that followed the Russian elections. For this topic, the news slowly shifted over the cause of days from a focus on the election itself to the steadily increasing demonstrations. This gradual shift towards a new topic is relatively difficult to detect for the approaches used in this study. The effective detection of these types of event may require a novel FSD approach that is not solely based on dissimilarity.

An inspection on the timeliness of the first stories detected reveals weaknesses in both approaches, and potentially an important aspect that should be taken into consideration in attempts to improve FSD. Timeliness of the detection is currently not addressed by the traditional evaluations that use a DET-curve and the tradeoff between recall and precision. To evaluate future work that addresses this issue, an additional metric to compare the timeliness of FSD approaches is required.

5.5. Conclusion

In this study, we propose a novel approach for the task of First Story Detection based on clustering news articles that are likely to be topically related, and estimating the novelty of newly formed clusters by comparison to previously seen news articles. We compared this approach to a baseline that estimates the novelty of a single news article by the cosine similarity to its nearest neighbor. The evaluation shows that the proposed model outperforms the existing baseline both in tradeoff between missed first stories and false positives, and in tradeoff between recall and precision. An analysis of the timeliness of the first story detections revealed that both systems missed early detection on some

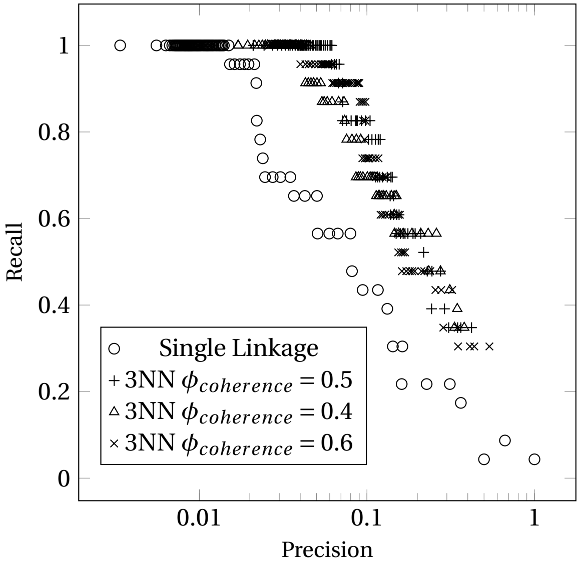


Figure 5.2: Plotted point show the recall/precision that correspond to the systems' effectiveness at the given threshold.

cases, and that there are specific cases such as evolving events that are particularly hard to detect.

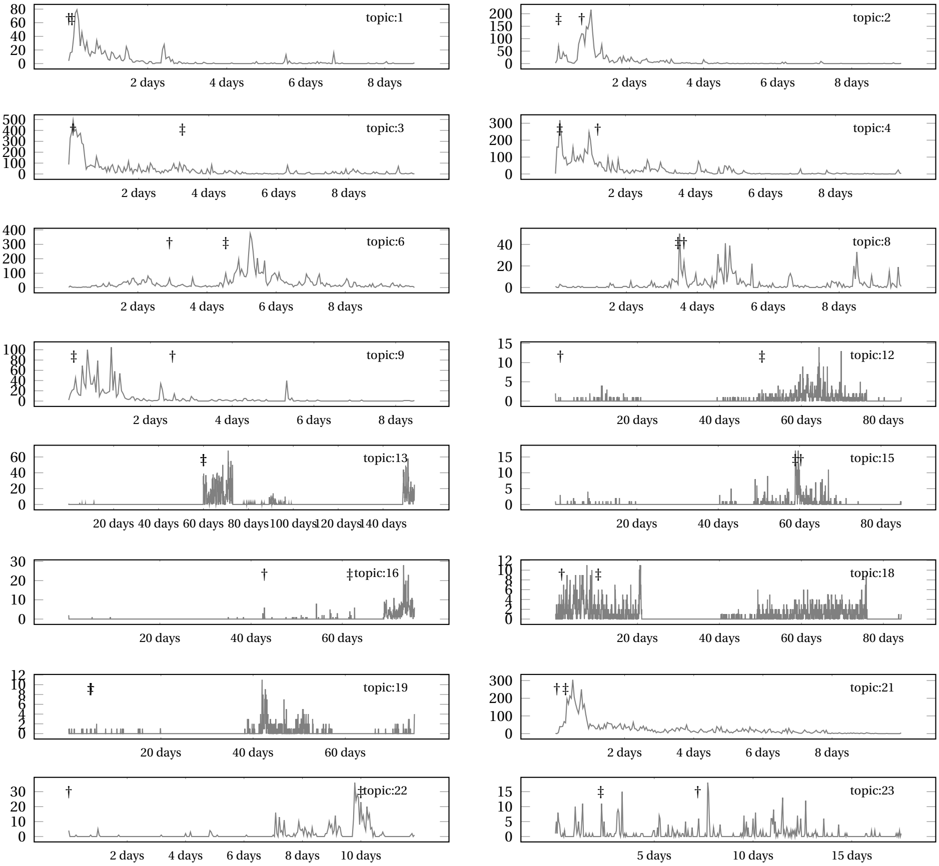


Figure 5.3: On the y-axis is the number of relevant news articles for the topic per hour, over time on the x-axis. A ‡ indicates when a first story is detected by 3NN $\phi_{coherence}$ = 0.5, and a † indicates when a first story is detected by Single Linkage, both at the ‘optimal’ novelty threshold that obtained recall=1 and the highest precision.

III

Proximity of semantic vectors for recommender systems

6

Exploring deep space: Learning personalized ranking in a semantic space

Recommender systems leverage both content and user interactions to generate recommendations that fit users' preferences. The recent surge of interest in deep learning presents new opportunities for exploiting these two sources of information. To recommend items we propose to first learn a user-independent high-dimensional semantic space in which items are positioned according to their substitutability, and then learn a user-specific transformation function to transform this space into a ranking according to the user's past preferences. An advantage of the proposed architecture is that it can be used to effectively recommend items using either content that describes the items or user-item ratings. We show that this approach significantly outperforms state-of-the-art recommender systems on the MovieLens 1M dataset.

Keywords: *Recommender Systems · Semantic Vectors · Neural Networks*

The contents of this chapter has been published as J.B.P. Vuurens, M.A. Larson, A.P. de Vries, In Proceedings of the RecSys Workshop on Deep Learning for Recommender Systems, ACM, 2016

6.1. Introduction

State-of-the-art collaborative-filtering systems recommend items by analyzing the history of user-item preferences. Alternatively, content-based systems analyze data about the items, and suggest items to a user that are most similar to the items she liked in the past. Past research has shown collaborative filtering to be more effective than content-based systems, however, it also has a few disadvantages over content-based models. Firstly, collaborative filtering requires a large quantity of user data to infer preference patterns between users. Secondly, these algorithms are generally considered less capable of recommending novel items, while novel items may be preferable over popular items for instance when a recommender system is repeatedly used to look for a job or a house (Fleder and Hosanagar, 2009; Lops et al., 2011). In cases when collaborative filtering is less applicable, content-based approaches can be used to complement the list of recommendations.

In recent years we have seen a rise in the use of semantic space models for various tasks such as translation and analogical reasoning (LeCun et al., 2015). In such a space, each element is represented as an abstract vector, which typically captures semantic properties of the elements and semantic relations between elements.

In this work, we present a novel approach for the recommendation of items, that first structures items in a semantic space and then for a given user learns a function to transform this space into a ranked list of recommendations that matches the user's preferences. We show that the same architecture can be used to effectively recommend items using either the text of user reviews or user-item ratings. We evaluate this approach using the MovieLens 1M dataset, and show that the proposed approach using user-item ratings significantly outperforms state-of-the-art recommender systems.

6.2. Semantic spaces for RecSys

6.2.1. Semantic spaces

Lowe (2001) defines a semantic space model as a way of representing the similarities between contexts in a Euclidean space. A semantic space represents the **intersubstitutability** of items in context, i.e. items may effectively be substituted by nearby items in a semantic space. This definition is based on an observation by Firth (1957): "you shall know a word by the company it keeps". The intuition for this distributional characterization of semantics is that whatever makes words similar or dissimilar in meaning, it must show up distributionally in the lexical company of the words.

When comparing highly-dimensional objects such as text documents, similarity measures are only reliable for nearly identical objects, since the "curse of dimensionality" makes dissimilar items appear equi-distant (Bengio et al., 2003; Beyer et al., 1999). In a semantic space, the curse of dimensionality can be counteracted by representing items using non-sparse vector elements that describe the strength of the association with item-related data. Various methods have been proposed to learn semantic representations. Landauer and Dumais (1997) perform a Latent Semantic Analysis by considering the informativeness of words in documents, i.e. word co-occurrences that are evenly distributed over documents are less informative than those that are concentrated in a small subset. Lowe and McDonald (2000) used a log-odds-ratio measure to explicitly factor out chance co-occurrences.

6.2.2. Towards recommendations

In this work, we propose to learn semantic item representations, for the task of recommending items to a user. The key idea is to position all items in a high-dimensional normalized semantic space, in such a way that items that are more likely to substitute each other are positioned closely together. Ideally, the items are positioned in such a way that for each user there is a region that exclusively contains items that the user (knowing or unknowingly) likes, making it possible to recommend items to a user by simply finding the best region in semantic space. The substitutability between items can be inferred from the observation of being jointly liked by a subset of users, or in a content-based setting by having similar descriptions.

To illustrate such a semantic space, Figure 6.1 shows a normalized t-SNE projection for movies in the MovieLens 1M dataset, representing every movie as a vector over the ratings by users. Using 2 dimensions, such a normalized space is shaped like the edge of a circle, on which the proximity between movies reflects their proximity to other movies in the user-item ratings matrix. For readability we show only the titles of the top-20 most popular movies after all 4000 movies were distributed over the available space. The three red clusters are movies that are positioned in close proximity, which we colored red and represented as a list for readability, e.g. a cluster with Star Wars IV and seven other movies. The distribution of the three red clusters over space indicates the existence of users that like movies in only one of these clusters. However, if we assume that there are also users that like the movies in two or even all three of these clusters, how can we construct a semantic space so that for every user an optimal region of interest exists? Using a normalized two dimensional space, there is no possible model that contains regions for all combinations of two out of three of these clusters without covering additional space. It requires a higher-dimensional space to create more overlapping regions for users with partially shared preferences.

In a near-optimal high-dimensional semantic space, the 'best' recommendation candidates are likely to be positioned in close proximity to the items the user rated highly. To recommend items to a specific user, we propose to find a function that transforms a semantic space into a one-dimensional space in which her rated items are ranked accordingly, reasoning that in the transformation the rated and unrated items that are of interest to the user will end up in a close to optimal position.

6.2.3. Related work

A tried-and-true approach for recommending items to a user is to learn latent factors which describe the observed preferences of users towards items. Some of the most successful recommendation methods use matrix factorization to represent users and items in a shared latent low-dimensional space. The prediction of whether a user will like an item is commonly estimated by the dot product between their latent representations (Koren et al., 2009). The two main disadvantages to the latent factors learned are that they are not easy to interpret and that it cannot generalize beyond rated items. Different from matrix factorization, in our approach we do not optimize shared latent factors to represent users and items, but rather predict the substitutability between items. When

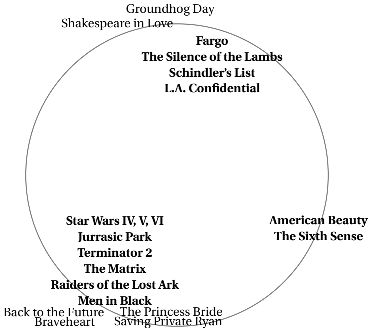


Figure 6.1: Example of a semantic space for the 20 most popular movies in MovieLens 1M. The figure is a normalized 2D t-SNE projection of the MovieLens user-item matrix. In bold are movies that are positioned very closely and therefore represented as a cluster.

the distance between vectors corresponds to their substitutability, the data can be interpreted more straightforward using the nearest neighbor heuristic and visualization techniques such as t-SNE. Visualization of latent factors is of interest to the recommender system community (cf. Németh et al., 2013). We also show that both user-item ratings and textual content can be used within the same framework, which makes it possible to generalize beyond rated items, however, we leave this for future work.

Collaborative Topic Regression (CTR) fits a model in latent topic space to explain both the observed ratings and the words in a document, where the topical distribution of documents is inferred using LDA (Wang and Blei, 2011). Dai et al. (2014) analyzed the difference between document representations that where generated by LDA and neural embeddings that were learned using the Paragraph Vector, and conclude that Paragraph Vectors significantly outperform LDA, although it is not clear why neural embeddings work better. Our model is similar to CTR in learning a model that is optimized to predict both ratings and content that is used to describe items; however, using a neural network we neither need to explicitly prescribe the type of data nor do we need to extract a topical model prior to learning the embeddings.

For item recommendation, pair-wise ranking approaches can be used the capture the pair-wise preferences over items. Baysian Personalized Ranking is a state-of-the-art approach that maximizes the likelihood of pair-wise preferences over observed and unobserved items (Rendle et al., 2009). However, Yao et al. (2015) argue that this approach cannot incorporate additional item metadata, and is difficult to tune on sparse data. They propose to use LDA to reduce dimensionality of the data to overcome those deficiencies. In this work, we also present a pair-wise ranking approach. The key difference lies in the structure of the learned semantic space, which is learned with a Paragraph Vector architecture, chosen with the goal of making regions of interest more easily separable when dealing with a large number of dimensions. In a sense, such a space resembles a metric space, meaning that our approach can be viewed as a proposal to learn a ranking function based on vector algebra rather than by estimated likelihood.

For the task of recommending movies, Musto et al. (2016) use semantic vectors for movies that are the average over the Word2Vec embeddings of the words on the movie's

Wikepedia page. In our approach the semantic vectors are learned to jointly predict observations for movies, rather than an average over the semantic vectors of individual words. For the recommendations, Musto et al. (2016) regard a user's preference as the average vector of their highly rated movies, and then movies are ranked according to their distance to this point in semantic space. In this work, instead of positioning the user in semantic space a function is learned that transforms the structure in semantic space into a ranking that is optimized for a user's past preferences.

For the task of personalizing relevant text content to users, Elkahky et al. (2015) propose a content-based approach to map users and items to a shared semantic space, and recommend items that have maximum similarity to a user in the mapped space. By jointly learning a space using features from clicked webpages, news articles, downloaded apps and viewed movie and TV program, they show that recommendations improve over those only learned over a single domain. Following the Deep Structured Semantic Model (DSSM) that was proposed by Huang et al. (2013), user and item features are mapped to 128-dimensional semantic vectors using a 5-layer architecture to maximizing the similarity between the semantic vectors of users and the items they interacted with in the past. In our work, a shallow neural network is used to learn item vectors that optimally predict their observed features using a shared weight matrix. To recommend items for a single user, the user-independent space is transformed according to her past preferences.

6.3. APPROACH

6.3.1. LEARNING SEMANTIC VECTORS

Bengio et al. (2003) propose to learn embeddings for words based on their surrounding words in natural language. Although the architecture that Bengio et al. proposed is still applicable for learning state-of-the-art semantic vectors, their approach received only moderate attention until Mikolov et al. (2013) used this idea to design highly efficient deep learning architectures for learning embeddings for words and short phrases, also known as Word2Vec. They show that the accuracy of the word embeddings increases with the amount of training data, and to some extent that the learning process consistently encodes some generalizations in the semantic vectors which can be used for analogous reasoning, such as the gender difference between otherwise equivalent words. This generalizing effect possibly occurs when a more efficient encoding can be used to jointly predict similar contexts for different words, although the exact conditions under which these generalizations are captured are not known. Recently, Le and Mikolov (2014) proposed an architecture to learn embeddings for paragraphs and documents.

In this study, semantic vectors for the items in a corpus are learned using the *Paragraph Vector* architecture described in Figure 6.2, which is similar to the PV-DBOW architecture proposed by Le and Mikolov (2014). The input (bottom) is a '1-hot lookup' vector, that contains as many nodes as there are items, and for every training sample only has the node that corresponds to the movie ID set to 1 while the other nodes are set to zero, which effectively looks up an embedding for a given movie m in weight matrix w_0 and places it in the hidden layer (middle). The output layer contains a node y for every possible observation in the training samples. The weight matrices w_0 and w_1

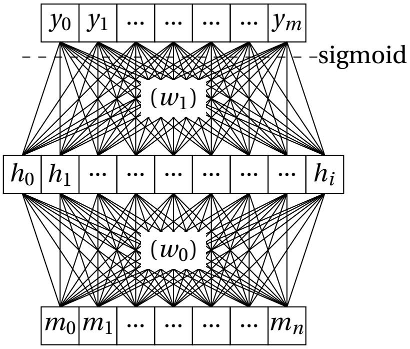


Figure 6.2: Deep learning architecture that is used to learn semantic vectors for items. The observations are streamed one-at-a-time, placing a movie-id in the input layer (bottom), which lookup a embedding in w_0 and places it in the hidden layer (middle). The model then updates weights w_1 of the observed item y and the embedding to optimize predictions using stochastic gradient descent.

respectively connect all possible input nodes, hidden nodes and output nodes. We learn the embeddings by predicting the outputs in a hierarchical softmax, i.e. all possible outputs are placed in a binary Huffman tree to learn the position of the observation in the tree rather than separate probabilities for each possible output (Mikolov et al., 2013). The item embeddings are learned together with a weight matrix w_1 by streaming over the observed features one-at-a-time in random order. For every movie, the network can generate a probability distribution over all possible observations by computing the dot product between the embedding with w_1. Using stochastic gradient descent, the embeddings and weights are updated to improve the prediction of the observed data. The learning process is similar to that described by Mikolov et al. (2013) for the learning of word distribution using a Skipgram architecture against a hierarchical softmax, except that no context window is used but rather all observations are predicted one-at-a-time.

To learn semantic vectors that capture the substitutability between items, the observations used to learn the semantic vectors should be representative for their substitutability. This can for instance be inferred from the observation that a group of users gave these items high ratings, but also from reviews that each describe an item or an opinion about the item. Lops et al. (2011) argue that existing content-based techniques require knowledge of the domain, however, learning item representations using a neural network has the advantage that patterns between items are learned automatically and therefore obviates the need for prior domain knowledge. In the evaluation, we will show that we can effectively learn semantic vectors for items using the same deep learning architecture on both user-item ratings as well as item descriptions.

The semantic vectors are learned from paired training samples (*item ID, observation*), where the observation can be an attribute of an item, a word that appear in an item's description (in this study a movie review), or an item's rating by a user. To learn semantic vectors from user-item ratings, in this study we corrected for the anchoring effects mentioned in Koren (2010) by interpreting the ratings as relative to its user's average; replacing ratings below the user's average with the label 'low' and ratings greater or equal to the average with the label 'high'. Additionally, we also learn unrated items with a label 'unrated', often being items that the user is not interested in. The input is transformed so that every observation becomes a single compound word, e.g. for Star Wars IV, which has id 240 in MovieLens 1M the rating 3 by user 73 (who has given an average rating of 3.4) is transformed into (240, 'user73_low'). Alternatively, in a content-based setting a review fragment for the same movie that contains "The masterpiece, the legend that made people..." is transformed into (240, 'the'), (240, 'masterpiece'), (240, 'the'), etc..

6.3.2. USER-SPECIFIC RANKING

In Section 6.2.2, we argued that for a near-optimal semantic space there should be a function that transforms this semantic space into a one-dimensional space in which a user's past preferences lie according to their ratings. In this work, we limited our search for such a function to finding a hyperplane for this transformation. Such a hyperplane is described by a vector that is orthogonal to the hyperplane, and the dot product with this vector projects the semantic vectors to a one-dimensional space according to their squared distance to the hyperplane, which is negative for items that lie on the opposite side of the hyperplane. By using a hyperplane, dimensions that are less useful for ranking the items can be down weighted or even ignored by choosing a hyperplane parallel to those dimensions.

To learn an optimal hyperplane, we propose a neural network architecture that optimizes the ranking over pair-wise preferences. Figure 6.3 shows a schematic of the architecture, which learns a hyperplane orthogonal to w_0 by stochastic gradient descent over pairs of item vectors a and b, given that item a has received a lower rating than b. The semantic vectors for a and b are not updated during learning. A shared weight matrix w_0 is used to compute a score of respectively r_a and r_b as the dot product between the semantic vectors and w_0. These scores are then combined using the fixed weights $(+1, -1)$, and filtered by a sigmoid function. The output layer directly provides the gradient $g \in [0, 1]$ that is used to update w_0, by subtracting $\eta \cdot g \cdot a$ from w_0 and adding $\eta \cdot g \cdot b$ to w_0. The learning rate η linearly descends from an initial value (in this study by default 0.025) to 0 during the learning process.

When estimating an optimal hyperplane to transform a semantic space into a ranking, we use all pairs of a movie b that was rating greater or equal to the user's average rating with a movie a that was rated lower than b or unrated. When learning the hyperplane, the system iterates ϕ_i-times over all item pairs that are rated differently by the user.

The time needed to learn the parameters of a hyperplane increases quadratically over the number of items the user has rated. Interestingly, there are several way to improve both the efficiency and effectiveness of the learning process. Koren (2010) observed that users' preferences change over time and shift between concepts. We hypothesize that simply using only the ϕ_t-most recently rated items may improve both the effectiveness and the efficiency of the recommender system. Another consideration for item recommendation is that optimally predicting the higher ranked items is more important than the ranking between lower ranked items. Typically, relatively few of the available items are of interest to the average user, and to avoid over-optimizing the prediction of unrated items over interesting items the unrated items can be down sampled. In this work, the down-sampling rate is controlled by a hyperparameter ϕ_d, e.g. when $\phi_i = 10$ itera-

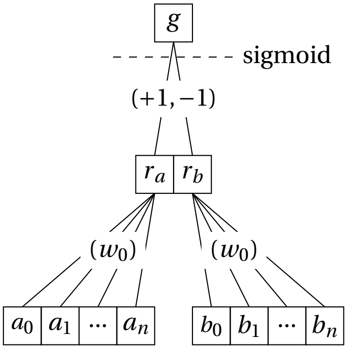


Figure 6.3: Neural network architecture that is used to learn the parameters w_0 of a hyperplane that optimally transforms items from an n-dimensional semantic space into a one-dimensional space, by optimizing the predicted order of pairs of item vectors a and b as rated by a user. The item pairs are streamed one-at-a-time, placing the semantic vector of the lower rated item in a and of the higher rated item in b. Starting with a random hyperplane w_0 the scores r_a, r_b are computed and the resulting gradient g is used to rotate the hyperplane towards a more optimal ranking using stochastic gradient descent.

tions are used with downsampling $\phi_d = 0.1$ every combination between a rated and an unrated item is used in exactly one randomly chosen iteration, while the combinations between two rated items are used ϕ_i times for learning.

6.4. EXPERIMENT

The proposed "Deep Space" approach (DS) first learns user-independent semantic vectors for items, which can then be transformed into a ranking that is optimized according to a single user's preferences. We will show that by using only the ϕ_t items the user rated prior to the time of recommendation, both efficiency and effectiveness are greatly improved. However, in order to evaluate the recommendation given for a timestamp without having used future information when learning the semantic vectors, the evaluation should proceed as in an online setting. Since our semantic space model is currently not-updatable, using an online evaluation strategy on the entire dataset is not feasible since for every item a new semantic space must be learned. To implement a fair, yet feasible, test procedure, we sampled a test-set from the dataset that consists of a user's temporarily latest ratings, then a single semantic space is learned using all ratings except those in the test set, and in the evaluation this model is used to predict the test samples. For this reason, the experimental systems use no information that lies in the future with respect to the target user at the moment of interaction with the test item.

In this paper, we carry out initial experiments that test the viability of the "Deep Space" approach. We chose MovieLens 1M because it is easily available and its properties are well-known, making it easy for others to understand and reproduce our findings. Note that we need a data set in which both ratings and reviews are available for the items. The MovieLens 1M dataset consists of 1 million ratings by 3952 users for 6040 movies on a 5-point scale. For the content-based experiments, we use the contents of the movies' user reviews on IMDB without their rating or username, and consider every word in the review text an observed word. To sample a validation and test set, we order

Table 6.1: Parameters tuned for MovieLens 1M

System	recall@10
BPRMF	$factors = 100$, $reg = 0.001$, $lrate = 0.025$, $iter = 30$
WMRF	$factors = 20$, $reg = 0.020$, $alpha = 0.1$, $iter = 10$
UserKNN	$k = 60$
DS-CB	$\phi_d = 1$, $\phi_t = 10$, $\phi_i = 10$
DS-VSM	$\phi_d = 20$, $\phi_t = 5$, $\phi_i = 10$
DS-CF	$\phi_d = 20$, $\phi_t = 5$, $\phi_i = 10$

the users by their number of ratings, and the ratings by the time they were submitted. Then, in that order of all ratings by all users, we mark every 25th rating. This ensures the test set matches the corpus' distribution over users rating volume, since prediction difficulty may be different between users that rated a few or many items. Then, if for a user n ratings are marked, from her temporarily-last n ratings the first half is assigned to the validation set, the last half to the test set, and in case of an odd number it is assigned to the shorter of the two sets or the validation set when equal in length. The models' parameters are tuned using the validation set, by training the model on all ratings except those in the test or the validation set. For the evaluation we use the test set after training the models on all ratings except those in the test set. All systems use the exact same training, validation and test set for the evaluation.

The effectiveness of the recommender systems is evaluated using recall@10 over the approximately 10k ratings in the test set that are a 4 or a 5 on a 5-point scale. The recall@10 metric is directly interpretable as the proportion of left-out items that a system returns in the top-10 recommendations.

6.5. RESULTS

We evaluate the effectiveness of our approach, by comparing the results of our approach to that of a popularity baseline and the MyMediaLite implementation of BPRMF (Rendle et al., 2009), WRMF (Hu et al., 2008), and UserKNN. The parameters for all models are tuned on the validation set that is described in Section 6.4, and the resulting parameters are shown in Table 6.1.

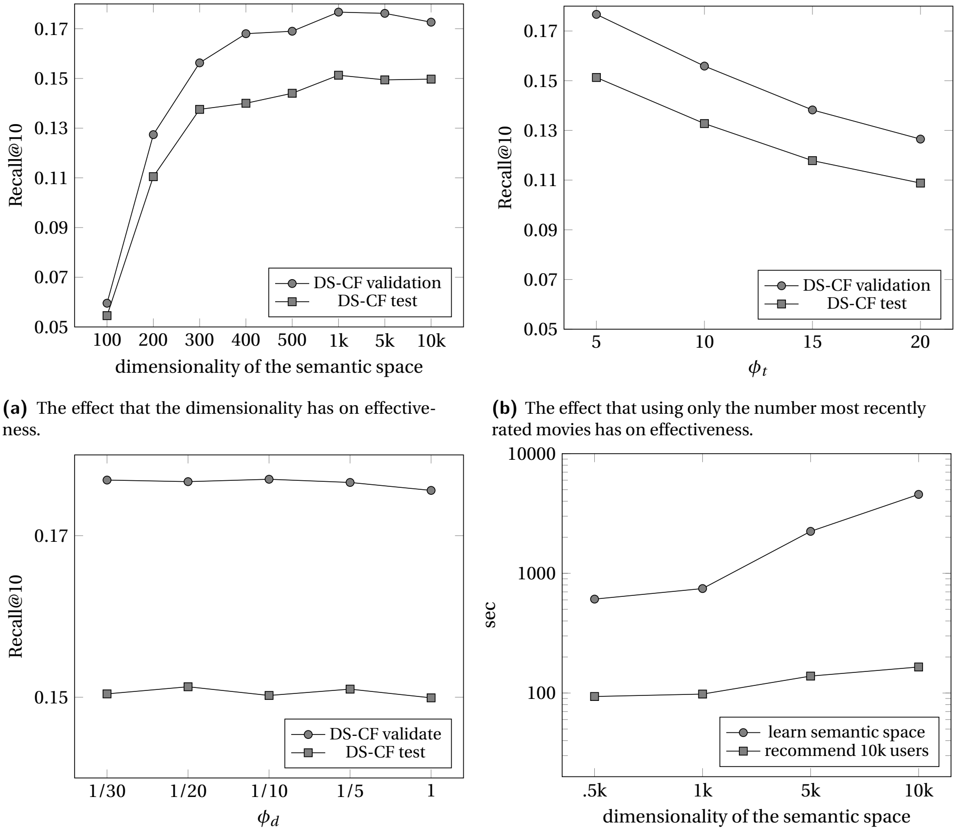
For the proposed model, we evaluate three variants: The DS-CB variant uses the Paragraph Vector to learn semantic vectors from the text of IMDB user reviews, and uses no rating information of other users than the user that is recommended to. The DS-VSM variant does not learn a contiguous semantic space using the Paragraph Vector, but uses a normalized vector space model (VSM) in which every user is a dimension and each item is represented as a vector consisting of its user ratings. The DS-CF variant uses the Paragraph Vector to learn a semantic space from the user-item ratings from which the recommendations are made. Table 6.2 reports recall@10 obtained by all models on the test set. We tested the differences between systems for statistical significance, using the McNemar test on a 2x2 contingency table of paired nominal results (a left-out item is retrieved in the top-10 of neither, one or both systems). In Table 6.2, all significant improvements have a *p-value* < 0.001. In these experiments, the DS-CF and DS-VSM

Table 6.2: Comparison of the effectiveness on MovieLens 1M. The subscripts in the column “sig. over” correspond to a significant improvement over the corresponding system, tested using McNemar test, 1-tailed, *p-value* < 0.001.

System	recall@10	sig. over
Pop	0.053	
BPRMF ^{1}	0.079	^{4}
UserKNN ^{2}	0.087	^{4}
WMRF ^{3}	0.089	^{4}
DS-CB-10k ^{4}	0.075	
DS-VSM ^{5}	0.119	^{1,2,3,4}
DS-CF-500	0.144	^{1,2,3,4,5}
DS-CF-1k	0.151	^{1,2,3,4,5}

models are significantly more effective than BPRMF, WRMF, UserKNN and DS-CB. By including the DS-VSM model in the evaluation, we show that the improvement is not only the result of learning semantic vectors with the Paragraph Vector, but is partially contributed by learning a hyperplane to optimally rank a user’s past ratings for the recommendation. However, since the DS-CF variant significantly outperforms the DS-VSM variant, we also show the benefit of learning semantic vectors with the Paragraph Vector which for generating recommendations is both more effective and more efficient. Although the representations learned with the Paragraph Vector are lower in dimensionality than the VSM over all users, typically, the DS-CF performs best in much higher-dimensional space than state-of-the-art matrix factorization approaches. The DS-CB variant that learns 10k dimensional semantic vectors from movie reviews is significantly less effective than the approaches that use user-item ratings. However, for items that have not been rated the content-based variant may provide an alternative.

We analyze the sensitivity of the hyperparameters ϕ_d, ϕ_t and the dimensionality of the semantic space. Hereto we perform a sweep over these parameters using the DS-CF model, changing only one hyperparameter at a time while setting the remaining two out of three parameters to $dimensionality = 1000$, $\phi_t = 5$, and $\phi_d = 20$. In Figure 6.3a, by changing the dimensionality of the semantic space we observe that the DS-CF model outperforms the VSM variant when dimensionality is at least 300, and that the effectiveness does not improve beyond the use of 1k dimensions. The degradation in performance when using less than 300 dimensions is possibly related to the linear transformation function that is used to rank the items, since in a lower dimensional space it may not be possible to position the items so that for all users there exists a linear function to generate a close to optimal ranking. In Figure 6.3b, we observe that using only the n most-recent ratings given by a user is more effective for lower values of n; when using more than five ratings to learn a transformation function the effectiveness degrades. In Figure 6.3c, shows the effect that down sampling of the used unrated items has on the effectiveness of learned transformation functions, where $\phi_d = 1$ equals no down sampling. In general, down sampling improves the efficiency of the recommendation while not having any negative impact on recall. This hyperparameter does not appear to be sensitive on this collection. The optimal value for these three hyperparameters may be



(a) The effect that the dimensionality has on effectiveness.

(b) The effect that using only the number most recently rated movies has on effectiveness.

(c) The effect that downsampling the use of unrated items has on effectiveness.

(d) The time to learn a semantic space using the Paragraph Vector on user-item ratings and the time to generate 10k recommendations by hyperplane projection.

Figure 6.4: The effect of hyperparameters on effectiveness and efficiency

6

collection dependent, and therefore need to be tuned.

We finally report about the efficiency of the proposed approach. All experiments were performed on a machine with two Intel(R) Xeon(R) CPU E5-2698 v3, which together have 32 physical cores. Using the test-set as described in Section 6.4, Figure 6.3d reports the wall time in seconds for learning a semantic space with the Paragraph Vector on the user-item ratings on the training data of the test-set, and the total time taken to generate a full ranked list for the approximately 10,000 items in the test set. For learning the semantic spaces, the user-item ratings were processed in 20 iterations, which for a 1000 dimensional semantic space takes 12.5 minutes. For the same dimensionality, the average time to rank all items using the parameter settings in Table 6.1 according to a user's preferences takes approximately 0.3 core seconds.

6.6. Conclusion

For the task of recommending items to a user, we propose to learn a semantic space in which substitutable items are positioned in close proximity. We show that these spaces can be learned from item reviews as well as user-item ratings, using the same deep learning architecture. To recommend items to a specific user, we learn a function that optimally transforms a user-independent semantic space into a ranking that is optimized according to the user's past ratings. In the experiments that use user-item ratings, this approach significantly outperformed BPRMF, WRMF and UserKNN on the MovieLens 1M dataset. When a semantic space is learned from user reviews on IMDB, the results are not as effective as these existing collaborative-filtering baselines, but may be useful to recommend novel items or when there is an insufficient amount of user-item ratings available to use collaborative filtering.

An interesting direction for future work is to extend function space to non-linear functions, that are potentially more optimal when the dimensionality of the semantic space is reduced. Another interesting direction is to jointly learn item representation based on content and collaborative filtering data, which may improve recommendation on sparse collections and for cold start cases.

7

Conclusion

In this thesis, proximity has been used as a guiding principle for the estimation of relevance for various tasks. By thorough analyses of the data on how proximity-based features can be useful for the problem at hand we managed to improve over state-of-the-art models, but there does not appear to be a general model of proximity that applies to all domains. In Part I of the thesis, we looked at the proximity of query terms in text as an indication of document relevance. In the second part we researched how the proximity of sentences that are published in news articles is indicative for their salience. Finally, in Part III, we investigated the representation of items in a semantic space and how those representations can be used for recommender systems. We will discuss the proximity effects found in each domain in more detail.

Q1: HOW IS PROXIMITY BETWEEN TERMS INDICATIVE FOR THE RELEVANCE OF A DOCUMENT?

For query-based document retrieval, a context-free retrieval model ranks the documents in a collection according to a query. As a basis, we use a state-of-the-art algorithm that considers terms as being independent, and analyzed how the distance between co-occurring query terms should be reflected in the relevance score of a document. Over a set of 350 queries on a news wire collection, we learned that the likelihood of two query terms to jointly appear in a document that is relevant decays by an inversely proportional function over the number of words that separate them. We propose a scoring function that adjusts the score that is assigned to individual query terms using the observed distance between jointly appearing query terms. The evaluation shows that the results are comparable to or better than the best competing model. Adding term dependency appears more effective for Web collections than for news wire, as predicted by Metzler and Croft (2005).

For modeling term dependency in a retrieval model, in previous works Tao and Zhai (2007) and Zhao and Yun (2009) have experimented with a range of convex functions which in our experiments do not generalize towards Web collections. The contribution of this work, is that we learned that the decay of the relevance score of a document can be described by an inversely proportional function, which does seem to generalize to Web collections.

Q2: HOW IS THE PROXIMITY BETWEEN PUBLISHED SENTENCES IN NEWS ARTICLES INDICATIVE OF THE SALIENCE AND NOVELTY OF NEWS?

For the detection and tracking of news topics, we propose a 3-nearest neighbor clustering approach (3NN) that effectively identifies salient information by clustering sentences that appear in new articles. Based on the observation that important news is often published by multiple newspapers within a short timespan, the 2-degenerate cores in a 3NN clustering graph are used to identify salient news. A sentence that is published is added to the summary when it is considered to be salient, it contains novel information and a better match to the most recently seen information about the topic than the sentences in the summary so far.

When there is much news regarding a topic, a generated summary may become long at the expense of its readability. By subsequently applying a hierarchical clustering method, the summary can be segmented into subtopics, each of which are more coherent. To assess whether sentences in different news articles discuss the same subtopic, the proximity between news articles is found to be most indicative of whether their contained sentences discuss the same subtopic. To improve the estimated distance between documents, we introduce a normalized information gain measure that appears to be more effective than the Cosine distance for this task.

For the discovery of new topics in the news stream, we apply 3NN on entire news articles using normalized information gain to estimate the similarity between the articles, and show there is a consistent improvement in precision over a single linkage first story detection baseline.

The main contribution of Part II is the 3NN clustering algorithm. The proximity mechanism that is used in 3NN extends the nearest neighbor heuristic; 2-degenerate cores in the clustering graph are formed in the absence of other sentences that are more similar for one of the core's sentences. By using the formation of 2-degenerate cores as criterium for the detection of salient sentences, a relative measure of proximity is used rather than an absolute measure of proximity that is parameterized by a threshold. By design, this approach has a higher latency than existing approaches, since detection is delayed until at least three sentences from different newspapers cluster together. However, compared to other approaches, the precision is significantly higher, which may be favorable for use in online systems when there is less urgency and a delay in news pushed is acceptable.

Q3: HOW CAN THE PROXIMITY BETWEEN SEMANTIC ITEM VECTORS BE USED AS THE BASIS OF ITEM RECOMMENDATION?

In Part III of this thesis we study the use of high-dimensional semantic item vectors as the basis of item recommendation. In our approach, the items are ranked according to the signed distance of their semantic vector to a hyperplane, using hyperplane coefficients that optimally rank a user's past preferences. The proposed framework can be used for either collaborative filtering or content-based recommendations. The collaborative filtering variant significantly outperforms state-of-the-art collaborative filtering baselines on Movielens1M, while the content-based variant is potentially useful to improve the recommendation of new or rarely rated items.

In a semantic space, the difference between two objects represents a meaning, e.g. a difference in movie genre, lead actors, suspense. Of all the factors that are encoded in the semantic space, a user may favor some factors while being indifferent to others. The ranking of items by their distance to a hyperplane allows to optimize for factors that a user cares about, while ignoring factors that the user is indifferent to. Therefore, by using hyperplane ranking on items that are represented in a semantic space, it does not matter if two items that are equally preferred by a user are separated by factors that user is indifferent to.

We systematically analyzed the differences between the collaborative filtering variants, to explain the difference in effectiveness. We included a variant in which we represented the items as a vector over their user ratings rather than learning item embeddings (DS-VSM), and rank the items using the proposed hyperplane ranking approach on these vectors. This variant significantly outperformed the collaborative filtering baselines we compared against, but is significantly less effective than our Deep Space Collaborative Filtering model (DS-CF). This experiment shows that there is a significant benefit in learning semantic vectors, but also that the gain of our collaborative filtering model over existing baselines is not fully explained by just the use of using semantic vectors.

In another experiment we learned latent item vectors with Bayesian Personalized Ranking (BPR) of the same dimensionality that is used for the DS-CF, and used these in the proposed hyperplane ranking approach instead of the embeddings that are learned with Paragraph2Vec. The results are comparable to that of DS-CF, showing a more general benefit of using semantic representations instead of vectors over user-item ratings, and that there is no significance difference between the algorithms to learn semantic representations.

Since BPR is also a pairwise learning-to-rank approach, the observed improvement seems to be the result of learning the semantic item representations using all available data, while using only the n-most recently rated items by the targeted user to rank the items. When using all available data to recommend items, the results of the Deep Space model are comparable to that of the UserKNN baseline. A final remark is that the improvement of DS-CF is only observed when learning semantic vectors of a much higher dimensionality than what is optimal for existing collaborative filtering approaches.

FUTURE WORK

In this Section we outline a number of areas for possible future work.

In the study on the proximity of terms in Chapter 1, we use a retrieval function that combines a term independence baseline with the proximity information of all term combinations. Although this retrieval function provides results that are comparable to or better than the best competing model, we observe that proximity can be counter effective in special cases of query term combinations. An interesting future direction of research is to analyze for which queries or term combinations proximity expansions are likely to improve results. For example, adjectives are more likely to be bound to a specific noun and it may be counter effective to combine adjectives with other nouns, e.g. for the query “white house rose garden” the user is more likely interested in documents containing “white house” and “rose garden” than in documents containing “white rose” and “garden house”. The selection of useful term combinations may be improved by using linguistic information or named entity detection. More insights to predict the useful term combinations (as well as those to ignore) may alleviate the negative effects of proximity as well as improve the runtime performance of the system.

In Chapter 2, for the summarization of news for a given topic, we assign a cluster of news articles to a topic when at least one of its news articles contains all the terms from the topic description in its title. This simple matching function is less suitable for elaborate topic descriptions or when the topic is likely to described using synonyms. An interesting direction for future work is to explore how clusters can be more effectively matched to topic descriptions.

In Chapter 4, we show that normalized information gain is more effective than cosine similarity to cluster news articles that discuss the same subtopic. The cosine is most strongly affected by (dis)similarities amongst the terms with the highest (IDF weighted) frequencies, while in contrast the entropy of documents is most strongly affected by terms that appear only once. Since terms occurrences in documents are Zipf distributed, normalized information gain uses a larger part of the vocabulary to estimate the similarity between documents. Possibly, an estimation of whether two documents discuss the same subtopic requires a comparison in more detail than an estimation of whether they discuss the same topic. Therefore, an interesting direction for future work is to test the hypothesis that normalized information gain is more suitable than cosine similarity for a detailed comparison of information.

In Chapter 6, for the task of recommending items to users, we propose a framework that can be used with collaborative filtering data and with text descriptions. The evaluation shows that collaborative filtering works significantly better than content-based recommendations on Movielens1M. However, a known problem with collaborative filtering is that it cannot be used to effectively recommend new or rarely rated items. An interesting direction for future work is to jointly learn item representations from both content descriptions and collaborative filtering data, that can be used to effectively recommend both cold start cases as well as sufficiently rated items.

ACKNOWLEDGEMENTS

ACKNOWLEDGMENT

I have many, many people to thank for making this happen. I will name some of them in chronological order, and apologize if I have left someone out by mistake.

First of all, to Ron Mantel for planting the seed; explaining that the Dutch government subsidized HBO staff members to acquire a PhD as part of a program to how to initiate practical research as part of the University program. To the very supportive Gert de Ruiter director of the Faculty IT & Design, who believed in my capabilities to succeed, and his persistence in persuading other parties to agree. To Ineke van der Meule of the Centrum for Lectoraten en Onderzoek (CLO) and Bert Mulder lector Informatie Techniek en Samenleving (ITS), for their advice and their consent. To Peter Mika, who probably does not recall recommending Arjen de Vries as a possible promotor. To Arjen, who I have come to appreciate as a friend, who has always been very open, encouraging, knowledgable and shown a very broad interest. Initially I was faced with a choice between the 'safety' of a familiar topic, or the bold choice of diving into information retrieval for which I had great interest but knew little about. Arjen made the choice very simple, and frankly, I would not have learned so much, and had such a pleasant and interesting adventure without him. To Carsten Eickhoff, who by giving me great feedback on my research and writing has been extremely helpful during the first year. Also a thanks to other past and present members of the Multimedia Information Retrieval group of TU Delft, most specifically to Raynor Vliegendhart, and Martha Larson for their interest in my research and helpful comments.

Starting from November 2012, the faculty of IT and Design, CLO and lectoraat ITS have generously sponsored the expenses for this PhD research which allowed me to work on this three days a week for four years, for which I am very grateful. Also to the Yahoo Faculty Research and Engagement Program, and especially to Peter Mika and Roi Blanco for giving me the opportunity to work on the Yahoo servers and the collaboration on the study on the summarization of news articles. Also a big thanks to the SARA/SURF foundation, who maintain the Dutch national e-infrastructure on which we implemented the models and run the experiments on large data volumes, and especially to Evert Lammerts, Jeroen Schot and Mathijs Kattenberg for providing awesome support.

I have been warned up front that a PhD may be more demanding and requires additional spare time. Indeed, it really has become like this new all consuming hobby that you do spend weekends on. So naturally, the greatest sponsor has been the love of my life Ilona, who had to endure my mood swings, and missed out on fun times together while I was glued to a screen, stuck with my head in a stack of papers or attending conferences.

Appendices

A

PROOF OF RANK EQUIVALENCE KLD AND QL

The KLD function presented by Zhai and Lafferty (2004) and Eq. 1.1 and the Query Likelihood function presented by Metzler and Croft (2005) and Eq. 1.10 are both Dirichlet smoothed language model estimates. Although these functions assign different scores to documents, they are in fact rank equivalent, as we will show here. In this proof, q_i is a term in query Q, μ is the Dirichlet prior parameter, $|D|$ is the number of words in document D, $tf_{q_i,D}$ is the number of times q_i appears in D, cf_{q_i} is the number of times q_i appears in collection C, $|C|$ is the total number of words in C, and $P(q_i|C)$ is the simple likelihood estimate that q_i appears in C.

$$
\begin{aligned}
QL(Q,D) = \quad & \sum_{q_i \in Q} \log\left(\left(1-\frac{\mu}{\mu+|D|}\right)\cdot\frac{tf_{q_i,D}}{|D|}+\frac{\mu}{\mu+|D|}\cdot\frac{cf_{q_i}}{|C|}\right) && \text{(A.1)}\\
= \quad & \sum_{q_i \in Q} \log\left(\frac{\cancel{|D|}}{\mu+|D|}\cdot\frac{tf_{q_i,D}}{\cancel{|D|}}+\frac{\mu}{\mu+|D|}\cdot\frac{cf_{q_i}}{|C|}\right) && \text{(A.2)}\\
= \quad & \sum_{q_i \in Q} \log\left(\frac{1}{\mu+|D|}\cdot\left(tf_{q_i,D}+\frac{\mu\cdot cf_{q_i}}{|C|}\right)\right) && \text{(A.3)}\\
= \quad & \sum_{q_i \in Q} \log\left(\frac{1}{\mu+|D|}\cdot\frac{1}{|C|}\cdot\left(tf_{q_i,D}\cdot|C|+\mu\cdot cf_{q_i}\right)\right) && \text{(A.4)}\\
= \quad & \sum_{q_i \in Q} \log\left(\frac{\mu}{\mu+|D|}\cdot\frac{cf_{q_i}}{|C|}\cdot\left(1+\frac{tf_{q_i,D}\cdot|C|}{\mu\cdot cf_{q_i}}\right)\right) && \text{(A.5)}\\
\overset{\text{rank}}{=} \quad & \sum_{q_i \in Q} \log\left(\frac{\mu}{\mu+|D|}\cdot\left(1+\frac{tf_{q_i,D}\cdot|C|}{\mu\cdot cf_{q_i}}\right)\right) && \text{(A.6)}\\
\overset{\text{rank}}{=} \quad & \sum_{q_i \in Q} \log\left(1+\frac{tf_{q_i,D}}{\mu\cdot Pq_i|C}\right)+|Q|\cdot\log\frac{\mu}{\mu+|D|} && \text{(A.7)}\\
= \quad & KLD(Q,D) && \text{(A.8)}
\end{aligned}
$$

In Equation A.5, we can eliminate the likelihood that the term appears in the corpus, which is the same for every document. Equation A.6 is thus rank equivalent to Equation A.5. In Equation A.7, the KLD function is derived by multiplying the document prior with a document independent constant, thus completing the proof.

B

RESULTS OF BEST TREC RUNS

In Table B.1, we present the highest StatMAP obtained by any system that participated at the TREC ad-hoc tracks. For the 2011 and 2012 Web Tracks, the best scoring systems outscore the proximity models by a large margin. The best TREC systems are complete retrieval systems, that filter out spam, use Learning To Rank on a range of features. In 2009 and 2013 there is less difference between the TREC best system and the proximity models. The first year these Web collections were used, no training data was given to the participants.

Table B.1: On the left, the highest MAP score obtained by any system that participated during the TREC ad-hoc task, and on the right the highest StatMAP score obtained by any system that participated for the Web Track ad-hoc tasks. The results were obtained by automatic systems using the topic title only, except for runs marked with a † which possibly used more information (e.g. topic description).

Test Set	MAP	
TREC1	n/a	
TREC2	0.3144	†
TREC3	0.4012	†
TREC5	0.2466	†
TREC6	0.2876	
TREC7	0.2614	
TREC8	0.3063	

Test Set	StatMAP
WT09	0.0855
WT11	0.2165
WT12	0.2168
WT13	0.1769

Curriculum Vitæ

Jeroen Bastiaan Pieter Vuurens

28-08-1968	Born in Amstelveen, The Netherlands.

Education

1993–2002	BSc Informatics and Information Science The Hague University of Applied Sciences
2003–2010	MSc Business and ICT Open University

Awards

2015	ECIR best reviewer award

LIST OF PUBLICATIONS

12. **Vuurens, J.B.P., Larson, M.A. and de Vries, A.P.**, *Exploring Deep Space: Learning Personalized Ranking in a Semantic Space*, In Proceedings of the RecSys Workshop on Deep Learning for Recommender Systems, ACM, 2016.

11. **Vuurens, J.B.P. and de Vries, A.P.**, *First Story Detection using Multiple Nearest Neighbors*, In Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval, pages 845–848, ACM, 2016.

10. **Vuurens, J.B.P., Eickhoff, C. and de Vries, A.P.**, *Efficient Parallel Learning of Word2Vec*, In Proceedings of the ICML Deep Learning Workshop, 2016.

9. **Vuurens, J.B.P., de Vries, A.P., Blanco, R. and Mika, P.**, *Hierarchy construction for news summarizations*, In Proceedings of the SIGIR TAIA Workshop, 2015.

8. **Vuurens, J.B.P. and de Vries, A.P.**, *CWI and TU Delft at the TREC 2015 Temporal Summarization Track*. In Proceedings of the Text REtrieval Conference, 2015.

7. **Vuurens, J.B.P., de Vries, A.P., Blanco, R. and Mika, P.**, *Online news tracking for ad-hoc information needs*. In Proceedings of the 2015 International Conference on The Theory of Information Retrieval, pages 221–230, ACM, 2015.

6. **Vuurens, J. B. P., de Vries, A. P., Blanco, R. and Mika, P.**, *Online news tracking for ad-hoc queries*. In Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, pages 1047–1048, ACM, 2015.

5. **Vuurens, J. B. P. and de Vries, A. P.**, *Distance matters! Cumulative proximity expansions for ranking documents*. Information retrieval, 17(4):380–406, Springer, 2014.

4. **Bellogín, A. and Gebremeskel, G. G. and He, J. and Lin, J. and Said, A. and Samar, T. and de Vries, A. P. and Vuurens, J. B. P.**, *CWI and TU Delft at TREC 2013: Contextual suggestion, federated web search, KBA, and web tracks*, In Proceedings of the Text REtrieval Conference, 2013.

3. **Vuurens, J. B. P. and de Vries, A. P.**, *Obtaining high-quality relevance judgments using crowdsourcing*, Internet Computing, 16(5):20–27, IEEE, 2012.

2. **Vuurens, J. B. P., Eickhoff, C. and de Vries, A. P.**, *Managing the Quality of Large- Scale Crowdsourcing*. In Proceedings of the Text REtrieval Conference, 2011.

1. **Vuurens, J. B. P., de Vries, A. P. and Eickhoff, C.**, *How much spam can you take? an analysis of crowdsourcing results to increase accuracy*, In Proceedings of ACM SIGIR Workshop on Crowdsourcing for Information Retrieval, pages 21–26, ACM, 2011.

BIBLIOGRAPHY

James Abello and François Queyroi. Fixed points of graph peeling. In *Advances in Social Networks Analysis and Mining (ASONAM), 2013 IEEE/ACM International Conference on*, pages 256–263. IEEE, 2013.

Charu C Aggarwal and S Yu Philip. On clustering massive text and categorical data streams. *Knowledge and information systems*, 24(2):171–196, 2010.

James Allan, Ron Papka, and Victor Lavrenko. On-line new event detection and tracking. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 37–45. ACM, 1998.

James Allan, Victor Lavrenko, Daniella Malin, and Russell Swan. Detections, bounds, and timelines: Umass and TDT-3. In *Proceedings of Topic Detection and Tracking Workshop (TDT-3)*, pages 167–174. Vienna, VA, 2000.

James Allan, Rahul Gupta, and Vikas Khandelwal. Temporal summaries of new topics. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 10–18. ACM, 2001.

Javed Aslam, Matthew Ekstrand-Abueg, Virgil Pavlu, Fernado Diaz, and Tetsuya Sakai. TREC 2013 Temporal Summarization. In *Proceedings of the 22nd Text Retrieval Conference (TREC), November*, 2013.

Gaurav Baruah, Adam Roegiest, and Mark D Smucker. The Effect of Expanding Relevance Judgements with Duplicates. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*. ACM, 2014.

Marcia J Bates. The design of browsing and berrypicking techniques for the online search interface. *Online review*, 13(5):407–424, 1989.

Doug Beeferman, Adam Berger, and John Lafferty. A model of lexical attraction and repulsion. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics*, pages 373–380. Association for Computational Linguistics, 1997.

Nicholas J Belkin, Robert N Oddy, and Helen M Brooks. Ask for information retrieval: Part i. background and theory. *Journal of documentation*, 38(2):61–71, 1982.

Michael Bendersky and W Bruce Croft. Modeling higher-order term dependencies in information retrieval using query hypergraphs. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, pages 941–950. ACM, 2012.

Michael Bendersky, Donald Metzler, and W Bruce Croft. Learning concept importance using a weighted dependence model. In *Proceedings of the third ACM international conference on Web search and data mining*, pages 31–40. ACM, 2010.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. A neural probabilistic language model. *Journal of Machine Learning Research*, 3:1137–1155, March 2003. ISSN 1532-4435.

Kevin Beyer, Jonathan Goldstein, Raghu Ramakrishnan, and Uri Shaft. When is nearest neighbor meaningful? In *Database Theory-ICDT'99*, pages 217–235. Springer, 1999.

Douglas Biber, Stig Johansson, Geoffrey Leech, Susan Conrad, Edward Finegan, and Randolph Quirk. *Longman grammar of spoken and written English*, volume 2. MIT Press, 1999.

Stefan Büttcher, Charles LA Clarke, and Brad Lushman. Term proximity scoring for adhoc retrieval on very large text collections. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 621–622. ACM, 2006.

Ben Carterette and James Allan. Research methodology in studies of assessor effort for information retrieval evaluation. In *Large Scale Semantic Access to Content (Text, Image, Video, and Sound)*, pages 738–757. CID, 2007.

Ben Carterette, Virgil Pavlu, Evangelos Kanoulas, Javed A Aslam, and James Allan. Evaluation over thousands of queries. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 651–658. ACM, 2008.

Pete Chapman, Julian Clinton, Randy Kerber, Thomas Khabaza, Thomas Reinartz, Colin Shearer, and Rudiger Wirth. CRISP-DM 1.0 step-by-step data mining guide. 2000.

Chun-Hung Cheng, Ada Waichee Fu, and Yi Zhang. Entropy-based subspace clustering for mining numerical data. In *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 84–93, 1999.

Hai Leong Chieu and Yoong Keok Lee. Query based event extraction along a timeline. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 425–432. ACM, 2004.

Charles LA Clarke, Gordon V Cormack, and Elizabeth A Tudhope. Relevance ranking for one to three term queries. *Information Processing & Management*, 36(2):291–311, 2000.

Kevyn Collins-Thompson and Jamie Callan. Estimation and use of uncertainty in pseudo-relevance feedback. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 303–310. ACM, 2007.

Thomas Cover and Peter Hart. Nearest neighbor pattern classification. *Information Theory, IEEE Transactions on*, 13(1):21–27, 1967.

W Bruce Croft, Howard R Turtle, and David D Lewis. The use of phrases and structured queries in information retrieval. In *Proceedings of the 14th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 32–45. ACM, 1991.

Ronan Cummins and Colm O'Riordan. Learning in a pairwise term-term proximity framework for information retrieval. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 251–258. ACM, 2009.

Andrew M Dai, Christopher Olah, and Quoc V Le. Document embedding with paragraph vectors. *NIPS 2014 Deep Learning and Representation Learning Workshop*, 2014.

Owen de Kretser and Alistair Moffat. Effective document presentation with a locality-based similarity heuristic. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 113–120. ACM, 1999.

Ap Dijksterhuis, Maarten W Bos, Loran F Nordgren, and Rick B Van Baaren. On making the right choice: The deliberation-without-attention effect. *Science*, 311(5763):1005–1007, 2006.

Ali Mamdouh Elkahky, Yang Song, and Xiaodong He. A multi-view deep learning approach for cross domain user modeling in recommendation systems. In *Proceedings of the 24th International World Wide Web Conference*, pages 278–288. ACM, 2015.

Günes Erkan and Dragomir R Radev. LexRank: Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research (JAIR)*, 22(1):457–479, 2004.

Joe Fagan. Automatic phrase indexing for document retrieval. In *Proceedings of the 10th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 91–101. ACM, 1987.

John R Firth. A synopsis of linguistic theory. 1957.

Jonathan G Fiscus and George R Doddington. Topic detection and tracking evaluation overview. In *Topic detection and tracking*, pages 17–31. Springer, 2002.

Daniel Fleder and Kartik Hosanagar. Blockbuster culture's next rise or fall: The impact of recommender systems on sales diversity. *Management science*, 55(5):697–712, 2009.

Evgeniy Gabrilovich, Susan Dumais, and Eric Horvitz. Newsjunkie: providing personalized newsfeeds via analysis of information novelty. In *Proceedings of the 13th international conference on World Wide Web*, pages 482–490. ACM, 2004.

Jianfeng Gao, Jian-Yun Nie, Guangyuan Wu, and Guihong Cao. Dependence language model for information retrieval. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 170–177. ACM, 2004.

Alon Halevy, Peter Norvig, and Fernando Pereira. The unreasonable effectiveness of data. *IEEE Intelligent Systems*, 24(2):8–12, 2009.

David Hawking and Paul Thistlewaite. Proximity operators-so near and yet so far. In *Proceedings of the 4th Text Retrieval Conference*, pages 131–143, 1995.

Ben He, Jimmy Xiangji Huang, and Xiaofeng Zhou. Modeling term proximity for probabilistic information retrieval models. *Information Sciences*, 181(14):3017–3031, 2011.

Yifan Hu, Yehuda Koren, and Chris Volinsky. Collaborative filtering for implicit feedback datasets. In *Proceedings of the 2008 Eighth IEEE International Conference on Data Mining*, pages 263–272. Ieee, 2008.

Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. Learning deep structured semantic models for web search using clickthrough data. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*, pages 2333–2338. ACM, 2013.

Bernard J Jansen, Amanda Spink, and Tefko Saracevic. Real life, real users, and real needs: a study and analysis of user queries on the web. *Information processing & management*, 36(2):207–227, 2000.

Margarita Karkali, François Rousseau, Alexandros Ntoulas, and Michalis Vazirgiannis. Efficient online novelty detection in news streams. In *Proceedings of the 14th international conference on Web Information Systems Engineering*, pages 57–71. Springer, 2013.

E Michael Keen. The use of term position devices in ranked output experiments. *Journal of Documentation*, 47(1):1–22, 1991.

Yehuda Koren. Collaborative filtering with temporal dynamics. *Communications of the ACM*, 53(4):89–97, 2010.

Yehuda Koren, Robert Bell, Chris Volinsky, et al. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.

Thomas K Landauer and Susan T Dumais. A solution to plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological review*, 104(2):211, 1997.

Bjornar Larsen and Chinatsu Aone. Fast and effective text mining using linear-time document clustering. In *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 16–22, 1999.

Victor Lavrenko. *A Generative Theory of Relevance.* PhD thesis, University of Massachusetts Amherst, 2004.

Victor Lavrenko and W Bruce Croft. Relevance based language models. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 120–127. ACM, 2001.

Quoc V Le and Tomas Mikolov. Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on Machine Learning*, pages 1188–1196. JMLR, 2014.

Gregor Leban, Blaz Fortuna, Janez Brank, and Marko Grobelnik. Event registry: learning about world events from news. In *Proceedings of the companion publication of the 23rd international conference on World wide web companion*, pages 107–110. International World Wide Web Conferences Steering Committee, 2014.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553): 436–444, 2015.

Qian Liu, Yue Liu, Dayong Wu, and Xueqi Cheng. ICTNET at temporal summarization track trec 2013. In *Proceedings of the The Twenty-Second Text REtrieval Conference*, 2013.

Tao Liu, Shengping Liu, Zheng Chen, and Wei-Ying Ma. An evaluation on feature selection for text clustering. In *Proceedings of the 20th International Conference on Machine Learning*, pages 488,495, 2003.

Xiaoyong Liu and W Bruce Croft. Passage retrieval based on language models. In *Proceedings of the eleventh international conference on Information and knowledge management*, pages 375–382. ACM, 2002.

Pasquale Lops, Marco De Gemmis, and Giovanni Semeraro. Content-based recommender systems: State of the art and trends. In *Recommender systems handbook*, pages 73–105. Springer, 2011.

Will Lowe. Towards a theory of semantic space. In *Proceedings of 23rd Annual Cognitive Science Society Conference*, pages 576–581. Lawrence Erlbaum Associates, 2001.

Will Lowe and Scott McDonald. The direct route: Mediated priming in semantic space. In *Proceedings of CogSci*, pages 675–680, 2000.

Yuanhua Lv and ChengXiang Zhai. Positional language models for information retrieval. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 299–306. ACM, 2009.

Christopher D Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to information retrieval.* Cambridge University Press, 2008.

Richard McCreadie, Craig Macdonald, Iadh Ounis, Miles Osborne, and Sasa Petrovic. Scalable distributed event detection for Twitter. In *2013 IEEE International Conference on Big Data*, pages 543–549. IEEE, 2013.

Polykarpos Meladianos, Giannis Nikolentzos, François Rousseau, Yannis Stavrakas, and Michalis Vazirgiannis. Degeneracy-based real-time sub-event detection in twitter stream. In *Ninth International AAAI Conference on Web and Social Media*, pages 248–257, 2015.

Claude Messner and Michaela Wänke. Unconscious information processing reduces information overload and increases product satisfaction. *Journal of Consumer Psychology*, 21(1):9–13, 2011.

Donald Metzler and W Bruce Croft. A Markov random field model for term dependencies. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 472–479. ACM, 2005.

Donald Metzler and W Bruce Croft. Latent concept expansion using markov random fields. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 311–318. ACM, 2007.

Jun Miao, Jimmy Xiangji Huang, and Zheng Ye. Proximity-based rocchio's model for pseudo relevance. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, pages 535–544. ACM, 2012.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26*, pages 3111–3119, 2013.

Cataldo Musto, Giovanni Semeraro, Marco de Gemmis, and Pasquale Lops. Learning word embeddings from wikipedia for content-based recommender systems. In *Proceedings of the 38th European Conference on Information Retrieval*, pages 729–734. Springer, 2016.

Ramesh Nallapati and James Allan. Capturing term dependencies using a language model based on sentence trees. In *Proceedings of the eleventh international conference on Information and knowledge management*, pages 383–390. ACM, 2002.

Bottyán Németh, Gábor Takács, István Pilászy, and Domonkos Tikk. Visualization of movie features in collaborative filtering. In *Proceedings of the 12th IEEE International Conference on Intelligent Software Methodologies, Tools and Techniques*, pages 229–233. IEEE, 2013.

Miles Osborne, Saša Petrovic, Richard McCreadie, Craig Macdonald, and Iadh Ounis. Bieber no more: First story detection using twitter and wikipedia. In *SIGIR 2012 Workshop on Temporal, Social and Spatially-aware Information Access*, 2012.

Ron Papka and James Allan. Topic detection and tracking: Event clustering as a basis for first story detection. In *Advances in Information Retrieval*, pages 97–126. Springer, 2002.

Saša Petrović, Miles Osborne, and Victor Lavrenko. Streaming first story detection with application to twitter. In *Human Language Technologies: The 2010 Annual Conference* *of the North American Chapter of the Association for Computational Linguistics*, pages 181–189. ACL, 2010.

Saša Petrović, Miles Osborne, and Victor Lavrenko. Using paraphrases for improving first story detection in news and twitter. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*, pages 338–346, 2012.

J. Ross Quinlan. Induction of decision trees. *Machine learning*, 1(1):81–106, 1986.

Dragomir Radev, Timothy Allison, Sasha Blair-Goldensohn, John Blitzer, Arda Celebi, Stanko Dimitrov, Elliott Drabek, Ali Hakim, Wai Lam, Danyu Liu, et al. MEAD-a platform for multidocument multilingual text summarization. In *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC 2004)*, 2004.

Dragomir Radev, Jahna Otterbacher, Adam Winkel, and Sasha Blair-Goldensohn. NewsInEssence: summarizing online news topics. *Communications of the ACM*, 48(10):95–98, 2005.

Yves Rasolofo and Jacques Savoy. Term proximity scoring for keyword-based retrieval systems. In *Advances in Information Retrieval*, pages 207–218. Springer, 2003.

Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. In *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence*, pages 452–461. AUAI Press, 2009.

Rishiraj Saha Roy, Anusha Suresh, Niloy Ganguly, and Monojit Choudhury. Improving document ranking for long queries with nested query segmentation. In *European Conference on Information Retrieval*, pages 775–781. Springer, 2016.

Tetsuya Sakai, Toshihiko Manabe, and Makoto Koyama. Flexible pseudo-relevance feedback via selective sampling. *ACM Transactions on Asian Language Information Processing (TALIP)*, 4(2):111–135, 2005.

Lixin Shi and Jian-Yun Nie. Using various term dependencies according to their utilities. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, pages 1493–1496. ACM, 2010.

Fei Song and W Bruce Croft. A general language model for information retrieval. In *Proceedings of the eighth international conference on Information and knowledge management*, pages 316–321. ACM, 1999.

Ruihua Song, Michael J Taylor, Ji-Rong Wen, Hsiao-Wuen Hon, and Yong Yu. Viewing term proximity from a different perspective. In *Advances in Information Retrieval*, pages 346–357. Springer, 2008.

Krysta M Svore, Pallika H Kanani, and Nazan Khan. How good is a span of terms?: exploiting proximity to improve web retrieval. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 154–161. ACM, 2010.

Tao Tao and ChengXiang Zhai. An exploration of proximity measures in information retrieval. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 295–302. ACM, 2007.

Stefanie Tellex, Boris Katz, Jimmy Lin, Aaron Fernandes, and Gregory Marton. Quantitative evaluation of passage retrieval algorithms for question answering. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 41–47. ACM, 2003.

Anastasios Tombros. *The effectiveness of query-based hierarchic clustering of documents for information retrieval.* PhD thesis, University of Glasgow, 2002.

Giang Binh Tran and Mohammad Alrifai. Indexing and analyzing Wikipedia's current events portal, the daily news summaries by the crowd. In *Proceedings of the companion publication of the 23rd international conference on World wide web companion*, pages 511–516, 2014.

Giang Binh Tran, Mohammad Alrifai, and Eelco Herder. Timeline summarization from relevant headlines. In *Proceedings of the IR research, 37th European conference on Advances in information retrieval*, pages 245–256, 2015.

Howard R Turtle and W Bruce Croft. Uncertainty in information retrieval systems. In *Uncertainty management in information systems*, pages 189–224. Springer, 1997.

Cornelis Joost Van Rijsbergen. A theoretical basis for the use of co-occurrence data in information retrieval. *Journal of documentation*, 33(2):106–119, 1977.

Cornelis Joost Van Rijsbergen. *Information Retrieval.* Butterworth, 1979.

Olga Vechtomova and Ying Wang. A study of the effect of term proximity on query expansion. *Journal of Information Science*, 32(4):324–333, 2006.

Jeroen B. P. Vuurens, Arjen P. de Vries, Roi Blanco, and Peter Mika. Online news tracking for ad-hoc queries. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval.* ACM, 2015a.

Jeroen B.P. Vuurens, Arjen P. de Vries, Roi Blanco, and Peter Mika. Hierarchy construction for news summarizations. In *SIGIR 2015 Workshop on Temporal, Social and Spatially-aware Information Access*, pages 1047–1048, 2015b.

Jeroen B.P. Vuurens, Arjen P. de Vries, Roi Blanco, and Peter Mika. Online news tracking for ad-hoc information needs. In *Proceedings of the 2015 International Conference on The Theory of Information Retrieval*, pages 221–230. ACM, 2015c. ISBN 978-1-4503-3833-2.

Chong Wang and David M Blei. Collaborative topic modeling for recommending scientific articles. In *Proceedings of the 22nd SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 448–456. ACM, 2011.

Yiming Yang, Tom Pierce, and Jaime Carbonell. A study of retrospective and on-line event detection. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 28–36. ACM, 1998.

Yiming Yang, Jian Zhang, Jaime Carbonell, and Chun Jin. Topic-conditioned novelty detection. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 688–693. ACM, 2002.

Weilong Yao, Jing He, Hua Wang, Yanchun Zhang, and Jie Cao. Collaborative topic ranking: Leveraging item meta-data for sparsity reduction. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, pages 374–380, 2015.

Hong Yu and Vasileios Hatzivassiloglou. Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences. In *Proceedings of the 2003 conference on Empirical methods in natural language processing*, pages 129–136. ACL, 2003.

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, 2004.

Chunyun Zhang, Weiyan Xu, Fanyu Meng, Hongyan li, Tu Wong, and Lixin Xu. The Information Extraction systems of PRIS at Temporal Summarization Track. In *Proceedings of the The Twenty-Second Text REtrieval Conference*, 2013.

Jinglei Zhao and Yeogirl Yun. A proximity language model for information retrieval. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 291–298. ACM, 2009.