Evaluation of Noisy Transcripts for Spoken Document Retrieval

Laurens van der Werff

PhD dissertation committee:

Chairman and Secretary:

Prof. dr. ir. A. J. Mouthaan, Universiteit Twente

Promotor:

Prof. dr. F. M. G. de Jong, Universiteit Twente

Members:

Prof. dr. D. K. J. Heylen, Universiteit Twente
Prof. dr. T. W. C. Huibers, Universiteit Twente
Dr. G. Jones, Dublin City University, Ireland
Prof. dr. ir. W. Kraaij, Radboud Universiteit Nijmegen
Dr. L. Lamel, Limsi - CNRS, Orsay, France
Prof. dr. ir. D. van Leeuwen, Radboud Universiteit Nijmegen

The research reported in this thesis was funded by the Netherlands Organization for Scientific Research (NWO) for the project CHoral - Access to oral history (grant number 640.002.502). CHoral is a project in the Continuous Access to Cultural Heritage Research (CATCH) Programme.

CTIT Ph.D. Thesis Series No. 11-224
Center for Telematics and Information Technology
P.O. Box 217, 7500AE
Enschede, The Netherlands.

SIKS Dissertation Series No. 2012-24
The research reported in this thesis was carried out under the auspices of SIKS, the Dutch Research School for Information and Knowledge Systems.

ISBN: 978-94-6203-066-4
ISSN: 1381-3617 (CTIT Ph.D. thesis Series No. 11-224)

Typeset with LaTeX. Printed by Wöhrmann Print Service.
Back cover Weighted Companion Cube graphic designed by Valve Corporation 2007.

EVALUATION OF NOISY TRANSCRIPTS

FOR SPOKEN DOCUMENT RETRIEVAL

DISSERTATION

to obtain
the degree of doctor at the University of Twente,
on the authority of the rector magnificus,
prof. dr. H. Brinksma,
on account of the decision of the graduation committee
to be publicly defended
on Thursday, July 5, 2012 at 16.45

by

Laurens van der Werff
born on March 1, 1975
in Leeuwarden, The Netherlands

Promotor: Prof. dr. F. M. G. de Jong

ISBN: 978-94-6203-066-4

voor Grijs

Acknowledgments

The writing of a PhD thesis is often a lonely and isolating experience, yet it is obviously not possible without the personal and practical support of numerous people. Thus my sincere gratitude goes to my mother, my sister, and my dear friends Auke, Jeroen, and Jo for their love, support, and patience over the last few years.

I thank my promotor Franciska de Jong for believing in me when I first applied for a position in the CHoral project, and for bringing me into contact with several well regarded scientists in the field I have been working in and actively seeking out and securing opportunities for internships. She has been instrumental in keeping me going all this time and has never been anything less than supportive and encouraging. She never gave up on me, even when the goal seemed to be slipping away. Not only was she my promotor, she also took on many supervision tasks and is the main reason for me being able to successfully complete this research.

Furthermore, I thank Willemijn Heeren for being my friend and colleague during the first years when we were stationed at the GemeenteArchief Rotterdam (GAR) together. Although our research often covered very different aspects of the project, she was always willing to lend an ear and provided me with many helpful clues on how to improve my work. Both my writing and presentation skills have improved dramatically thanks to her involvement. I also extend my gratitude to all other colleagues at the GAR, especially Jantje Steenhuis for actively supporting this project from the very beginning.

I have very fond memories of my two internships, and I thank both Lori Lamel and Gareth Jones for their hospitality. At Limsi in 2008/2009, Lori welcomed me at the TLP group and introduced me to many great people. Specifically, my office mates François Yvon and Nadège Thorez, and Tanel, Josep, Bac, Rena, Marc, Jean-Luc, Cecile, Guillaume, Martine, Ruth, and Megan. In Ireland at DCU in 2010 I found an equally good place at Gareth's lab where I also met a fantastic group of researchers with whom I watched most of the World Cup Association (hi John!) Football matches. Thanks for welcoming me into your group, Maria, Ágnes, Özlem, Robert, Jennifer, Sarah, John, Debasis, Ankit, Johannes, and all the others!

It would have been impossible to complete my research without the many fruitful discussions with friends and colleagues. Wessel Kraaij and Gareth Jones were of great help in improving some of my publications. Claudia Hauff was kind enough to patiently teach me much about IR research and selflessly shared her encyclopedic knowledge of relevant earlier work whenever I got stuck. I was able to bounce many ideas off of Marijn Huijbregts, who continued to help me out long after his leaving HMI. In addition, I thank the HMI group at the University of Twente for their support. CHoral member Thijs Verschoor has made several showcases possible and helped me out numerous times in imposing my will on my computer. HMI's loyal secretaries Charlotte Bijron and Alice Vissers have made my life so much easier by helping me find my way through many forms and regulations, and by generally being super nice and supportive. Furthermore, I

thank Hendri Hondorp for allowing me to avoid the helpdesk as often as I have been able to.

Finally, I thank NWO for starting the CATCH - Continuous Access To Cultural Heritage program, which funded the work in this thesis as part of the CHoral - Access to Oral History project. This project has not only funded my research, but has also provided me with a platform for interaction with other researchers and non-scientific entities from the world of cultural heritage.

Laurens

Contents

1

Introduction

The domain of the research reported in this thesis is Spoken Document Retrieval (SDR), which is generally taken to mean searching spoken word content. It implies matching a user information need as expressed in a textual query and the content of spoken documents, and ordering the results in order of expected relevance. Simply put, it means searching in speech in a way that is similar to web-search. In its simplest form, an SDR system can be implemented as shown in Figure 1.1. An Automatic Speech Recognition (ASR) system is used to produce a textual representation of a speech collection. This 'transcript' of the speech is used as the basis for an Information Retrieval (IR) task. For various reasons, automatic speech transcripts are bound to contain errors which may subsequently cause a bias in the results of the SDR task. For example, if a query term is erroneously omitted from a transcript, then the affected document may be ranked lower in a search session than without this 'deletion'. This thesis proposes and investigates a methodology for evaluating speech transcripts not just for the amount of noise but also for its impact on the results of the information retrieval component.

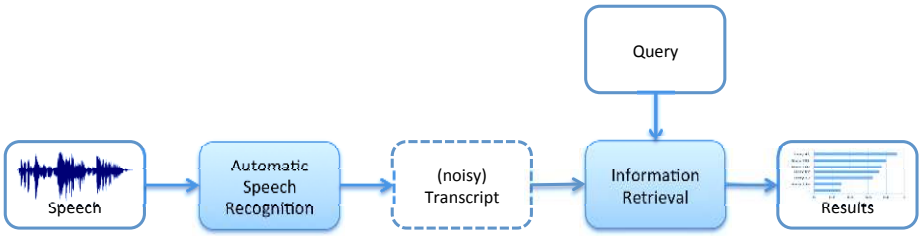

Figure 1.1: *An overview of SDR with the system shown as a simple concatenation of ASR and IR.*

1.1 Searching Spoken Content

Until the late middle ages, information was passed on orally or in handwritten form. The invention of the printing press opened up information to people in such a radical way that its inventor Johannes Gutenberg was elected in one poll as the most influential person of the millennium [Gottlieb et al., 1998]. In a sense, the internet is an extension of the printing press in that it allows for the publication of ideas, but without many of the practical barriers that were prevalent in the olden days (at least in the Western world). As the amount of information increased, as it first did in libraries and later on the internet, the desire emerged for some kind of structured access in order to find the information that is relevant for a specific need.

The initial solution before the invention and wide-spread use of computers was to use a system of manually assigned keywords for finding books within a library, often combined with a local index to further refine the search to pages or sections within a book. As (textual) information was digitized, automatic indexation became possible, paving the way for Information Retrieval as a topic for scientific research. Manual indexation is quite different from automatic indexation though: the former assumes decisions regarding relevance are made during the indexation stage leading to a selective index, whereas the latter is typically anticipating a retrieval stage where relevance is determined based on a complete index. Search engines such as Google or Yahoo! are implementations of best practices developed through scientific research in the field of IR and they are at least partially responsible for the abundant use of the internet as an information source to the general public.

More recently, the introduction of video-sharing portals such as YouTube and Vimeo have extended the possibilities for publishing and sharing by providing hosting opportunities for audiovisual (av) information. From an information retrieval perspective, access to this type of content often means a throwback to the days of manually assigned keywords. Full automatic indexation of audiovisual data in a manner that enables textual search is still an unsolved issue. Currently, the most reliable way of finding relevant fragments in this type of collection is through the use of manually assigned tags [Marlow et al., 2006], or by using contextual information from comments or referring sites. In practice, the additional effect of collaborative tagging [Peters and Becker, 2009] on popular av-sharing communities makes disclosure of the most popular content relatively straightforward.

The main difference with a library context however, is that in libraries all incoming books are treated with a similar amount of attention, typically by people with knowledge and experience of requirements of the tagging process. Also, all incoming content has often been explicitly selected for inclusion, whereas internet content is typically a mishmash of content of highly variable quality. As av-sharing portals depend largely on user-generated tags, an unintentional bias may be introduced in this manner into a tag-based retrieval system: more popular content is likely to have better quality tags, and is therefore more likely

to be found, and therefore more likely to receive additional tags. This makes it desirable to have an automatic indexation mechanism working alongside a keyword-based index in order to detect and potentially (manually) correct such biases.

Older speech collections, such as interview collections or radio archives are typically completely untagged. Retroactively adding such tags is often not feasible due to the sheer amount of speech that would have to be processed by hand. For example, in the context of the CHoral[1] research project, the radio archives of Radio Rijnmond were analyzed for automatic disclosure. As the largest Dutch regional radio station in the Rotterdam area, its archives span more than 20,000 hours of Dutch speech. All broadcast audio was archived and labeled for broadcasting date, but no additional metadata was ever kept or created for this collection. Despite being a potential treasure chest for historians interested in the area and its people, the collection has mostly remained unused. This is quite typical of (large) speech collections all over the world, especially in the domain of cultural heritage. Without some kind of automated indexation system, access to this type of collection is extremely limited.

Cases such as the Radio Rijnmond collection illustrate the potential for an automatic indexing solution for speech collections. Once a speech collection is stored in a computer-readable manner, and (computing) resources are available, an SDR solution can be engineered. The typical approach is to automatically generate a literal orthographic transcript of the speech using a Large Vocabulary Continuous Speech Recognition (LVCSR) system. The resulting transcript can then be treated as any other textual source and searched using Information Retrieval technology in order to retrieve and play relevant fragments. The usability of such systems is often thought of as inferior to text search, despite collaborative, large-scale investigations having shown that this need not be the case for English language broadcast news speech collections [Garofolo et al., 2000b].

One of the reasons for the expected difference in performance is thought to be the quality of the automatic transcript. English language studio-produced broadcast news speech is an almost ideal case for automatic transcription, and the number of errors in state-of-the-art systems for this type of speech can be well below 10%. Most popular IR approaches are expected to be robust enough to remain quite usable at this level of transcript noise. However, if the type of speech, recording, or spoken language is not ideal, transcript noise can rise rather quickly. For example, pilot experiments on the Radio Rijnmond collection, containing a mix of rehearsed and spontaneous speech under various conditions, indicated that transcript error rates exceeded 50%, much worse than the 20% error rate that was typically achieved by this system on broadcast news speech. In such conditions IR performance is expected to be reduced, with the most affected documents potentially becoming impossible to find.

In order to enable optimal access to non-broadcast-news type speech collections, it is therefore essential that retrieval bias that results from transcript noise

[1] http://hmi.ewi.utwente.nl/choral/

is recognized and avoided whenever possible. Any approach to the evaluation of ASR transcripts for SDR purposes must include the consequences of errors on the performance of the system as a whole. This can be achieved by evaluating the effectiveness of the IR system, but this typically requires a large amount of human-made resources, see Section 2.2.2. For most collections, these resources cannot be generated and the effect of transcript errors on SDR performance then remains unknown. Optimization of ASR system performance for a specific collection and/or expected information need is therefore currently unpracticable for many potentially valuable speech collections.

Disclosure of speech collections should not be restricted to academic environments, and not to collections for which large amounts of human resources can be expended. ASR systems provide ample opportunities for performance optimization, but evaluation of transcripts has so far been either unsuitable in the context of spoken document retrieval or hugely impractical. Implementing and optimizing ASR for a collection and information need should be achievable using off-the-shelf tools and without requiring a large amount of human-generated, collection-specific reference material. Our aim is to develop an evaluation methodology that enables an analysis of the quality of ASR transcripts which is both relevant in the context of spoken document retrieval and can be implemented without the need for large amounts of additional resources.

1.2 Problem Statement

Simple information retrieval systems count the frequency of terms and the frequency of documents that contain these terms to determine the potential relevance of a text for a query. Despite being somewhat basic, this approach yields quite usable results, but it also contains an inherent bias towards longer documents [Robertson and Walker, 1994]. Bias in search results is more or less a given, as neutrality is virtually impossible to define in this context. More advanced search mechanism such as used by Google or Bing try to avoid unintentional biases by using techniques such as Pagerank [Page et al., 1999] and personalization of results to intentionally introduce a different bias. An important challenge in IR is ensuring that biases that result from technical deficiencies, for example transcript noise in SDR, are properly recognized and where possible managed.

In the case of SDR, it is reasonable to assume that segments of speech which were transcribed rather poorly are likely to be ranked lower in comparison to what would result from their true content. Speech recognition errors therefore typically generate a negative bias for the correct retrieval of these segments. Poor speech recognition performance is often caused by noisy conditions (e.g., street interviews), accented speech (e.g., non-american or non-native english), or because of a mismatch in language use (e.g., the use of Named Entities or technical terms not present in the ASR lexicon). Such conditions are neither rare nor is it acceptable that they result in retrieval bias, as from a content point of view the affected fragments may be just as valuable as any non-accented, clean,

studio produced speech.

Evaluation of ASR transcripts is typically done using a count of errors, expressed as the Word Error Rate (WER). When optimizing an ASR system with the aim of reducing WER, it makes sense to first target the most frequent terms and the most common accent. Although a lexicon of only 10 unique terms (e.g., *the*, *to*, *of*, *a* ,*and*, *in*, *is*, *that*, *for*, and *are*) can cover more than 20% of all words that are spoken, it cannot express 20% of the meaning. The task of ASR in an SDR context is to somehow turn the content of speech into a form that is usable for a search engine, which is not necessarily equivalent to producing a literal orthographic transcript. In order to achieve maximum overall performance, one needs to have an evaluation mechanism that is capable of reflecting this task.

At this point, it is important to make a distinction between intrinsic and extrinsic evaluation methods. The former use data as-is and only evaluate inherent quality. A typical example is traditional ASR evaluation, which counts the number of errors at the level of words. Extrinsic evaluation methods go beyond the superficial characteristics of the outcome and measure the consequences of errors for the overall quality of a system and the way it performs a task. An example of this is IR evaluation, where errors are not measured directly but only for their impact on the ability to rank relevant before non-relevant documents.

Extrinsic evaluation of ASR transcripts in an SDR context can be achieved using an (intrinsic) evaluation of the retrieval results of the SDR system. The most popular method for determining the quality of retrieval results in a benchmarking context is Mean Average Precision (MAP). Although this measure is primarily suitable for comparing retrieval systems, there is no reason to assume that it cannot be equally useful for comparing transcript quality. Comparisons between IR performance on a ground-truth reference transcript and on an automatic transcript of speech, for example by using *relative* MAP, can be used to detect biases that result from transcript noise.

To calculate MAP, one needs relevance judgments for all stories in the results of an IR task (more on this in Chapter 2). With result lists typically containing around 1000 stories, and a simple evaluation needing at least 25 queries, this is rarely feasible. As a consequence, extrinsic evaluation of speech transcripts is practically impossible if we are limited to using MAP. We feel there is a need for a novel extrinsic evaluation paradigm, specifically one that can be used by anybody able to use an ASR system and requiring no more resources than for calculation of WER.

1.3 A Novel Approach to ASR Evaluation in an SDR context

In this section we first discuss how ASR-for-SDR was evaluated in benchmark conditions up until now, and why this is unfeasible for many ad-hoc collections.

We then propose an alternative which is as easy to implement as traditional ASR evaluation. The validity of the new approach for our application depends on the correlation between the new and old approaches. If the new approach results in values that have a high linear correlation with the traditional approach under a wide range of conditions, then one can reasonably assume that the two approaches measure the same thing. If the two approaches result in the same ranking of systems under a wide range of conditions, then they can be thought of as functionally equivalent for scenarios where system ranking is the main target.

1.3.1 TREC-style ASR-for-SDR Evaluation

Evaluation of SDR systems has traditionally focussed on challenges related to automatic transcription as the rest of an SDR system's tasks is largely similar to textual IR configurations that have been researched extensively in the context of the various Text REtrieval Conference (TREC) benchmarks, see Section 2.2.4. A basic evaluation of automatic transcript quality is a standard procedure when deploying an ASR system on a new task. It involves making a word-by-word comparison between a ground-truth reference and the automatically produced hypothesis transcript. To evaluate SDR, and specifically the impact of using speech rather than textual sources, the process is much more complex; an overview is shown in Figure 1.2.

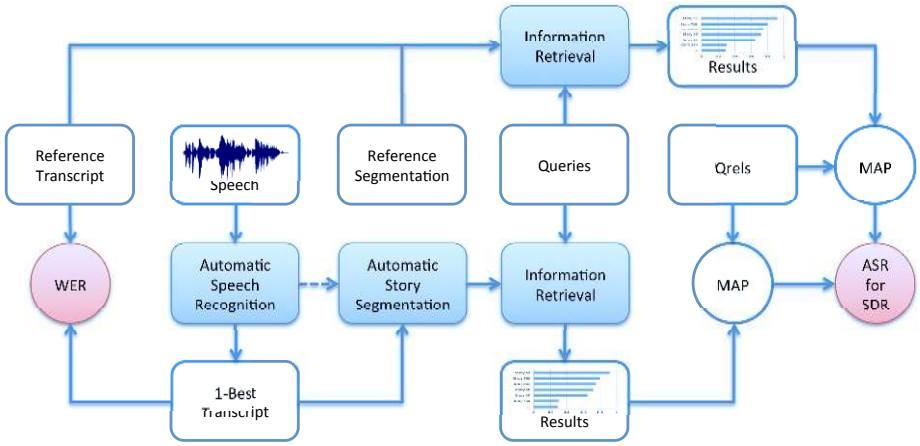

Figure 1.2: *A schematic overview of ASR-for-SDR evaluation using relative MAP as quality measure.*

In Figure 1.2, the red circles indicate two manners of evaluating the ASR process: WER and 'ASR-for-SDR'. The former is the standard intrinsic ASR evaluation that is typical for dictation-type systems, and the latter is an extrinsic measure in which MAP is compared for IR on a reference and on an automatic transcript. MAP is one of the most popular IR evaluation methods, however, its absolute value is highly dependent on the collection and the

queries used. In practice MAP is mostly used to rank systems, meaning only relative performance is established. For ASR-for-SDR evaluation it therefore makes sense to characterize the impact of transcript noise also by relative MAP, in this case for the difference between a noisy transcript and a ground truth reference.

A complete SDR system can be thought of as a black box that takes speech and queries as its input, and produces a ranked list of relevant fragments as its output. Such a system would require automatic speech recognition, automatic story segmentation, and information retrieval as its main components. Both ASR and IR solutions are readily available in off-the-shelf versions, for example Sphinx [Lee et al., 1990] and Nuance/Dragon[2] for speech recognition and Lemur[3] and Lucene[4] for information retrieval. Automatic story segmentation has not been investigated very extensively in this context, but an algorithm such as TextTiling [Hearst, 1994] has been shown to provide workable results, and implementations are freely available for various programming languages.

In addition to these functional components needed for performing the SDR task, extrinsic evaluation of ASR transcripts in an SDR context requires a number of resources. For basic ASR transcript evaluation, only spoken word content and a corresponding ground-truth reference transcript are required. In an SDR context, additional resources must be provided for extrinsic evaluation using MAP: a (reference) segmentation of the spoken content into retrieval units, topics/queries that are appropriate for the transcribed part of the collection, and relevance judgments for each query on every retrieval unit (qrels). Of these, qrels are usually the hardest to come by, as they are based on subjective judgements by humans [Voorhees, 2000]. For a small collection of only one thousand documents and 25 queries, up to 25,000 individual human-made judgments may be needed. Common practice in the creation of qrels is to only judge documents that are produced by various baseline retrieval systems, reducing the challenge slightly. However, for research purposes on SDR, some flexibility in the choice of topics/queries may be needed, something that is severely impaired by the laborious task of qrel creation. An additional caveat is that ideally, relevance judgments should be made on audio content rather than a transcript, as sometimes relevance may depend on (typically untranscribed) affect or non-verbal aspects of the recording. Outside of large benchmark settings, it is rarely feasible to generate a workable set of qrels for realistically sized textual collections, for speech collections it is likely to be even harder.

Performing an extrinsic evaluation of transcripts in this a manner is a rather unattractive scenario for someone developing an ASR system. A person optimizing an ASR system for use on a particular collection is typically expected to provide the resources needed for WER calculation. As SDR requires retrieval units, a transcript must typically be segmented into coherent stories. Additionally providing a reference segmentation implies only a limited amount of extra effort, and using off-the-shelf solutions for story segmentation and IR is also

[2] http://www.nuance.com
[3] http://www.lemurproject.org
[4] http://lucene.apache.org

quite feasible. However, creating queries and corresponding qrels is extremely time-consuming. As such, the 'ASR-for-SDR' evaluation that was done in the TREC SDR benchmarks and is shown in Figure 1.2 is simply not realistic for ad-hoc ASR system development.

1.3.2 Easy ASR-for-SDR Evaluation

The main reason why a TREC-style approach to ASR-for-SDR evaluation is unattractive for most practical scenarios is the need for qrels and the potential size requirement of the manually transcribed reference. Segmentation into coherent stories should be relatively easy to integrate into the manual transcription procedure, but generating queries is not so straightforward. The developer of an SDR system is not necessarily a user as well, and unless the collection has been manually transcribed in full, queries must be targeted towards the available portion of a collection. Our aim is to provide a framework for extrinsic evaluation of ASR-for-SDR which does not require any additional resources beyond what is typically used in traditional ASR evaluation, i.e., it should work with only a literal reference transcript of a small portion of the collection. In order to allow for the type of detailed analysis that is possible when using traditional extrinsic evaluation, the option should exist to manually provide additional resources in the form of story segmentation and queries, however a need for qrels must be avoided at all cost due to their inherent expense.

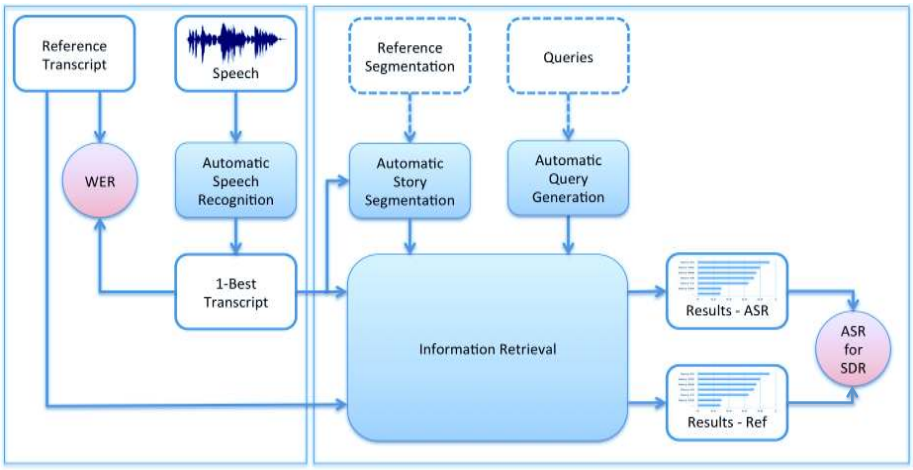

Figure 1.3: *Overview of a proposed novel framework for to ASR-for-SDR evaluation without the use of qrels or a need for manually generated queries and story boundaries.*

Our proposal for such a system is shown in Figure 1.3. The left side of the schematic shows a traditional ASR evaluation process, whereas the right side can be used as a 'black box' for ASR-for-SDR evaluation. The dashed lines indicate optional resources. The functional elements in the right part can be

implemented using off-the-shelf solutions. The main advantage of this approach is that qrels are no longer needed as MAP is not used anymore.

Having the right hand side of Figure 1.3 function in a fully autonomous manner requires, in addition to an IR system, automatic story segmentation and automatic query generation modules. We also need to process the difference between the ranked results from IR on a reference transcript and IR on an automatic transcript, in such a way that the resulting ASR-for-SDR measure is highly correlated with relative MAP. Furthermore the proposed evaluation model should work with a similar amount of manual transcripts as traditional ASR evaluation with WER. Only then can the approach as shown in Figure 1.3 be used as a functional replacement for the one in Figure 1.2.

1.4 Research Questions

The evaluation platform for ASR-for-SDR from Section 1.3.2 requires an IR system, and three other main components: i. automatic story segmentation, ii. comparing ranked results lists, and iii. automatic query generation. Several potential solutions for each of these can be found in the literature, but we need to establish which approaches work best in the context of the proposed system. We need to determine which method of comparing ranked results lists results in the highest correlation between MAP and ASR-for-SDR and how we may generate queries for such an evaluation. In addition we need to determine the amount of transcripts, and the number of topics/queries needed for reliable evaluation.

The design and implementation of each of the components is presented in a separate chapter. In this section we give an overview of the research questions that we aim to address in this thesis.

1.4.1 Automatic Story Segmentation

An automatic story segmentation task was researched as part of the Topic Detection and Tracking (TDT) benchmark [Fiscus and Doddington, 2002]. The implementation that was most popular in that context required the use of manually labeled training material for the probabilistically motivated algorithms to learn from. As our aim is to provide a method that can be used in isolation without any additional resources, the approaches that were used in the TDT segmentation task are typically unsuitable. In addition, the TDT segmentation evaluation relied on a cost-function, which is an intrinsic measure. Our need is for a segmentation system that works well in the context of an SDR system, of which we can only be sure if we do extrinsic evaluation.

We discuss several methods of automatic story segmentation that can be used without any collection-specific additional resources. The performance of these methods must be tested in an SDR context, so we use relative MAP to compare between automatically generated (artificial) and human-made (reference) boundaries. The research questions that we aim to answer are:

- Does extrinsic rather than intrinsic evaluation of artificial story boundaries lead to different conclusions with regards to the best choice of segmentation method?
- Which is the best method for automatic story segmentation without using additional resources in the context of SDR, based on extrinsic evaluation?
- What is the impact of artificial story boundaries on retrieval performance of an SDR system when compared to a reference segmentation?

1.4.2 Speech Transcript Evaluation for SDR

The core of the evaluation process that we proposed in Section 1.3.2 is the comparison between ranked results lists as produced by IR on a reference and an automatic transcript. This process is an extrinsic evaluation for the transcript, i.e., it measures the impact of the ASR noise on the results of the entire SDR process. Alternatives to WER that have been proposed so far were intrinsic and only showed some partial dependence on the IR system, for example by including some of the IR preprocessing or by focussing on terms with a higher expected importance, such as Named Entities.

Our aim is to establish methods for fully extrinsic evaluation that have high correlation with relative MAP. In addition, we investigate intrinsic approaches that are potential alternatives for WER, for example if the fully extrinsic approaches provide unsatisfactory results. The research questions that we investigate are:

- Can we evaluate ASR transcripts in an intrinsic manner that is more appropriate for SDR than traditional WER?
- Which method for extrinsic evaluation provides the highest correlation with relative MAP?
- Can extrinsic evaluation of ASR transcripts without qrels be reliably used to predict relative MAP?

1.4.3 Artificial Queries and Transcript Duration

Manually creating queries without qrels for ASR-for-SDR evaluation is expected to be quite feasible and possibly also quite desirable, as this enables one to focus specifically on the type of information requests that are expected to occur most frequently in the use of the system. However, if queries cannot be generated by actual users of the system, an alternative may be found in automatically generated queries. A reasonable approach might be to follow patterns that can be learned from other, well-studied, systems such as those found in TREC benchmarks. We implement an automatic query generation algorithm and test whether it results in ASR-for-SDR performance that is similar to using real queries. In addition we examine the amount of artificial queries that is required for reliably estimating ASR-for-SDR performance.

One of the concerns with using relative MAP for extrinsic ASR transcript evaluations, besides its reliance on qrels, is that one may need more reference material than for WER to get a meaningful result. MAP is calculated from qrels, which are a binary division of the collection into relevant and non-relevant stories. If the collection is very small, this division may be too coarse for getting an accurate estimate of MAP. A direct comparison of ranked results shouldn't have this problem, but may still require more resources than needed to calculate WER. It is important that the demands on the amount of manual transcripts do not limit the use of the extrinsic measures. We shall therefore examine how the ASR-for-SDR measures respond to the amount of reference transcripts that is available. As this requires experiments on many different subsets of the full collection, we use artificial queries. We formulate the following research questions:

- How many (artificial) queries are needed to reliably estimate ASR-for-SDR performance?
- Which method for automatic query generation results in the highest correlation between ASR-for-SDR measures and MAP as calculated from real queries?
- How is the reliability of the ASR-for-SDR performance measures affected by the duration of the manually transcribed reference speech?

1.5 Organization

Although this thesis is intended to be a single work, to be read in a linear manner, we also wanted to make sure that the various chapters are comprehendible on their own. As a result, there is some repetition in the argumentation and description of the test collection. We attempt to keep these to a minimum and refer to earlier chapters/sections as needed.

This thesis is organized as follows: A basic overview of Automatic Speech Recognition and Information Retrieval is provided in Chapter 2. It is intended to serve as an introduction for readers without a background in these fields, and provides explanations of the basic concepts that are used in this thesis. The first set of research questions, concerning automatic story boundary generation is investigated and answered in Chapter 3. Various methods of intrinsic and extrinsic ASR evaluation methods are proposed and examined in Chapter 4 in order to answer the second set of research questions. Automatic query generation and the requirements on the amount of references needed are dealt with in Chapter 5 along with answers to the third set of research questions. A short summary and conclusion are provided in Chapter 6.

2
Background

Implementing a Spoken Document Retrieval system involves a combination (or concatenation) of automatic speech recognition and information retrieval. As a result, this domain of research has received interest from both fields, as was demonstrated in the largest SDR benchmark so far (TREC SDR, 1997-2000). Some speech-oriented groups generated their own transcripts (Limsi, Cambridge University, Sheffield University), whereas other groups (AT&T, CMU) used the transcripts provided by TREC and focused on maximizing performance from their own flavor of retrieval engine.

Because the interests of readers of this thesis may be quite different, coming from either an IR or ASR research agenda, we cannot expect them to be intimately familiar with all important concepts from both fields. This chapter provides introductions to ASR and IR which should make the rest of this thesis more accessible for readers unfamiliar with both or either of these fields of research. A general overview of the workings of the ASR and IR approaches on which we build in our own research is included, but it is in no way exhaustive. For most of the methods we use, alternatives are available. We have however tried to use methods that are exemplary for the rest of the field and that represent the most popular approach in past benchmarks. The focus in this chapter is on aspects that return later in this thesis and have implications for SDR as we implemented it.

This chapter presents an overview of the workings of an Automatic Speech Recognition system in Section 2.1 and some basic information on popular approaches to Information Retrieval in Section 2.2. The chapter ends with Section 2.3 which includes some reflections on issues that arise when the two fields are combined. As this chapter contains no new research results, it can safely be skipped by anyone who is already sufficiently familiar with these subjects.

2.1 Automatic Speech Recognition

The transformation of speech into text has been a subject of research almost since the invention of computers, but has only started to improve significantly since the mid-1970s as computing power became sufficient for doing meaningful experiments within an acceptable time frame. Statistical modeling of speech

signals has been the basis of most ASR research, meaning that it was dependent on the availability of labeled data sets for training. The development of corpora that could be used as training material was an essential component in generating the performance improvements needed to make ASR a practical proposition. The National Institute of Science and Technology[1] (NIST) and the Linguistic Data Consortium[2] (LDC) have been instrumental in the creation, annotation, and distribution of speech and language resources for the scientific community. In addition, several tools have become available that could be deployed for scientific research purposes, including the Hidden Markov Model Toolkit (HTK) [Young et al., 2006] and Sphinx [Lee et al., 1990] speech recognition systems, and the Stanford Research Institute Language Modeling toolkit (SRILM) [Stolcke, 2002]. Together these tools enable building a complete basic speech recognition system without having to develop additional resources or perform low-level programming.

The remainder of this section provides an overview of the most important concepts in the automatic speech recognition process and its evaluation. We are will not cover functionalities such as knowledge representation (e.g., MFCC [Davis and Mermelstein, 1980], PLP [Hermansky, 1990]), speech signal normalization (e.g., CMS [Rosenberg et al., 1994], RASTA [Hermansky and Morgan, 1994], VTLN [Eide and Gish, 1996]), or adaptation (e.g., MAP [Gauvain and Lee, 1994], MLLR [Leggetter and Woodland, 1995], eigenvoices [Kuhn et al., 1998]) as these are mostly of interest for improving the state-of-the art in speech recognition and are not needed per se to understand the typical challenges for SDR research. Subjects that are included in this section are the basic speech recognition process (Section 2.1.1), evaluation and performance (Section 2.1.2), transcription alternatives (Section 2.1.3), and confidence scoring (Section 2.1.4).

2.1.1 Implementation

Due to the complexity of automatic speech recognition, even a proper overview of only the fundamentals would require more space than can reasonably be accommodated in the context of this thesis, but such overviews can be found in many other publications, e.g., [Rabiner and Juang, 1993, Young et al., 2006]. For readers unfamiliar with ASR in general, this section provides an introduction to some of the components in the most commonly used and most successful speech recognition systems. Figure 2.1 provides a high-level overview of the general process based on its functional components.

Audio can be represented either in the time domain, as a pressure wave, or in the frequency domain, composed of several distinct frequency components each with its own phase. Both representations are mathematically interchangeable, but the latter is much more convenient for statistical modeling due to the noisy phase component being separated from informational content: the levels of the frequency components, which can be used for spectral analysis. The frequency

[1] http://www.nist.gov
[2] http://www.ldc.upenn.edu

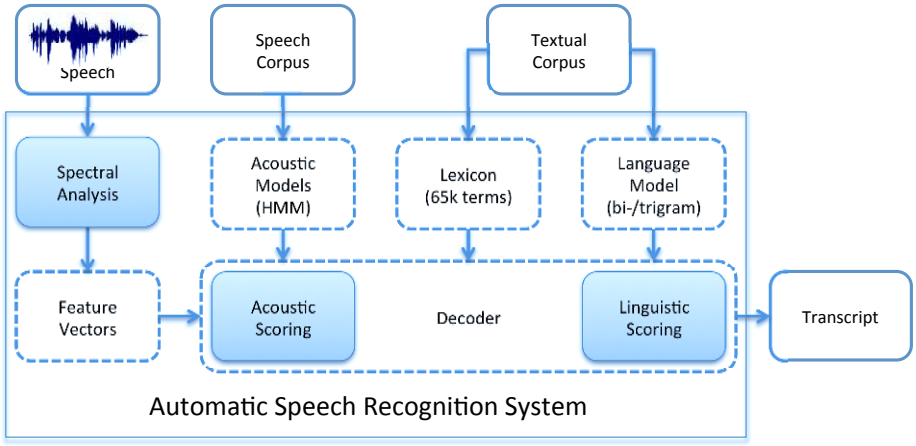

Figure 2.1: *Overview of an automatic speech recognition system. Acoustic data is processed into feature vectors; these can be matched to trained models and the most likely sequence of terms is produced as transcript.*

domain representation undergoes a lossy conversion into a stream of feature vectors, using windows of typically 25ms length, also called frames. Each vector contains the levels of 12 frequency bands plus an overall energy level. The frequency bands are usually based on a non-linear filter-bank [Burget and Hermansky, 2001], to ensure a frequency resolution that is similar to the sensitivities of the human ear. To optimize the information content for the statistical modeling of each band, the bands undergo a basic decorrelation [Ahmed et al., 1974] process. The 13 feature values are then augmented with delta- and delta-delta-components which model the difference with previous windows, resulting in a vector of 39 feature values per frame.

Speech is a concatenation of words, and words are built from individual sounds. The smallest component of speech that signifies a difference in meaning is called a phoneme. Which phonemes must be distinguished depends on the language. For English around 45 different basic phonemes are typically sufficient [Rabiner and Juang, 2003]. Using a (large) corpus of speech frames that are labeled for phoneme, one can capture statistical properties of such phonemes in acoustic models using Hidden Markov Models (HMM) [Young et al., 2006]. HMM's are a cornerstone of most ASR systems, being capable of capturing statistics on both feature values and time distortions. One can use HMM's to produce a probabilistic match between incoming speech frames and phonemes.

Choosing the most likely sequence of phonemes results in a phonetic transcript of the speech. Although there are circumstances where this is the desired output, the typically high error rate of such a transcript means that it is rarely suitable for dictation-type applications. An additional modeling layer is therefore added which contains a lexicon to limit the allowed phoneme sequences to known words, and language models which boost the likelihoods of word se-

quences that were previously found to naturally occur in use of the language.

Word sequences are scored using bi-, tri-, or fourgram language model likelihoods [Manning and Schutze, 1999]. The primary task of language models is to provide likelihoods for the co-occurrence of terms. Language models contain information about the a priori likelihood of a word occurring, i.e., *where* is a more frequent term than *lair*, but also about the conditional likelihood of a term, i.e., *the lair of the dragon* is a more likely phrase than *the where of the drag on*, despite the a priori likelihoods of the individual terms in the latter phrase typically being higher than those in the first one. The task of the decoder is to provide likelihoods for each possible combination of terms, given the feature vectors. But as the number of combinations for most practical applications is prohibitively high [Renals and Hochberg, 1996], the task is usually reduced to only providing, or rather finding, the most likely transcript. Alternatives to a literal transcript can take the shape of an n-best list, containing the top-n most likely transcripts, or a representation of the considered search space as a lattice structure, see Section 2.1.3.

An ASR decoder stage combines likelihoods as obtained from models of phonemes (acoustic models), a lexicon, and a model of the grammar (language models) into an overall likelihood score for a potential transcription of an utterance. The models are typically task-specific, so when a system is used for transcription of English language telephone conversations, one would use bandwidth-limited acoustic models of English phonemes. Further specialization can take place by using gender-specific models, or using models of accented speech, or even speaker-specific models when available and beneficial.

A large lexicon, e.g., one that contains all words that can reasonably be expected to occur in the language, seems attractive on the surface, but increases the potential search space of a transcription system. Furthermore, it may be difficult to correctly estimate the language model likelihoods of all these terms. A larger lexicon contains an increasing amount of rare terms, and since these terms are typically automatically learned from textual corpora, they may simply be misspellings. In extreme cases, this may result in an increase in the number of transcription errors, despite a reduction in Out-Of-Vocabulary (OOV) rate – words that were uttered but not in the lexicon. A high quality 'traditional' dictionary contains over a 170k entries[3], not including named entities. But one can achieve an OOV-rate of less than 2 per cent using a lexicon of around 65k (normalized) terms, which is a typical lexicon size for an English language ASR system, including named entities. State-of-the-art ASR systems sometimes use lexicons that contain 500k terms when the transcript is required to have proper capitalization, or for languages that have many compound words [Despres et al., 2009].

Bigram language models represent the likelihood of term B occurring after term A has been observed, trigrams extend the context to the two previous terms and fourgrams to the previous three terms. Given the exponential growth in the number of possible sequences, language modeling for ASR is typically

[3] http://oxforddictionaries.com

limited to trigrams. Using a general model of the language is already quite helpful in reducing the error rate of a transcript, but when a collection is on a specific subject, for example legal matters, World War II, or business meetings, there is potential for performance improvements through the use of task-specific language models. These are then typically 'mixed' [Clarkson and Robinson, 1997] with a general model as language models usually benefit tremendously from having an abundance of training material for estimation of likelihoods [Brants et al., 2007].

From the perspective of SDR, it is tempting to view speech recognition as a black box which simply converts the audio samples into a computer-readable textual representation. But as we just explained, there are many parameters and models involved, all of which can and should be tuned towards a specific task. In the case of SDR, this means that models must be chosen or adapted to reflect the type of speech and language use that is found in the collection. In addition, it may be necessary to optimize for the expected information requests, for example by including all terms in the lexicon that are expected to be used in queries.

2.1.2 Evaluation & Performance

The standard measure for automatic speech transcript quality is Word Error Rate (WER). It is calculated using the minimum Levenshtein distance [Levenshtein, 1966] between a reference (ground truth) transcript and a hypothesis, with alignment done at the word level. This alignment results in differences showing up as insertions (I), where a term was hypothesized but no equivalent term was found in the reference, deletions (D) which are the opposite, and substitutions (S), where one term was erroneously transcribed as another. The sum of insertions, deletions, and substitutions is divided by the total number of terms in the reference transcript (N) to produce WER, see Equation 2.1. Word Error Rate can be interpreted as the relative number of alterations that needs to be made (by a human) to an automatic transcript in order to correct it. Optimizing for WER is standard practice in speech transcription system development, especially for dictation-type tasks, where transcript noise may need to be removed retroactively. This is a relatively costly process as it must be done manually.

$$WER = \frac{I + D + S}{N} \times 100\% \tag{2.1}$$

For the ASR application that is central in this thesis, spoken document retrieval, the requirements are only superficially similar to those of a dictation task. In contrast to dictation-type applications, transcript errors are typically not corrected, as speech collections that are disclosed using SDR technology are usually of a size that requires bulk processing of the audio with only a minimum amount of human intervention. Errors therefore do not impact the time needed for post-processing, something that seems reasonably well addressed by WER, but they do impact the performance of the retrieval component of the

SDR system. In such situations, not the amount of errors determines retrieval performance, but rather the way these errors influence the search results. Word error rate offers a limited perspective on transcript quality, one which is mostly targeted towards traditional uses of ASR technology.

For NIST speech recognition benchmarks, WER has been the de facto measure of transcript quality. Figure 2.2 shows how WER has progressed in the best systems participating in the various NIST speech recognition benchmarks between 1988 and 2009. The various colors/markings represent different types of speech and different languages. More recent benchmarks have not only tried to pose bigger challenges for the participating systems, resulting in lower (initial) performance, but provided a platform for further development and performance gains. It is clear to see how 'Read Speech', and targeted applications such as 'Air Travel Planning Kiosk Speech' result in much better performance than 'Conversational Speech' and 'Meeting Speech'.

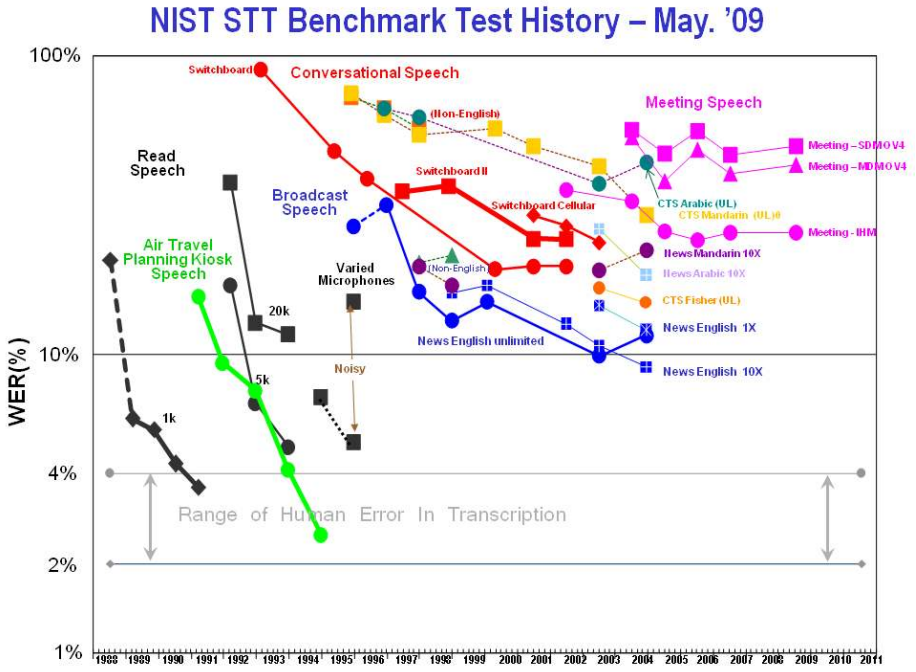

Figure 2.2: *Historic performance in terms of WER of speech recognition in official NIST benchmarks from 1988 to 2009*

As Figure 2.2[4] clearly shows, the expected performance of the ASR component is highly dependent on the type of speech that is in the collection. In the TREC SDR benchmarks, investigating performance of spoken document retrieval systems, only broadcast news speech collections were used. As such,

[4]http://www.itl.nist.gov/iad/mig/publications/ASRhistory/index.html

the quality of those transcripts was relatively high, in fact, performance was so good as to lead to the matter of ASR for SR on broadcast news famously being declared solved [Garofolo et al., 2000b]. For many other types of collections this may not be the case though, as ASR clearly struggles with several types of speech, e.g. non-scripted and conversational speech.

As WER was suspected to be suboptimal in the context of SDR, Term Error Rate (TER) [Johnson et al., 1999] was suggested as an alternative in the course of the TREC7 SDR benchmarks [Garofolo et al., 1998]. Information retrieval technology usually treats a collection of textual documents as 'bags-of-words', i.e., only the number of occurrences of words is considered, not their order. If word order is of no importance in an IR system, then it can also be ignored in evaluations of ASR transcripts for use in SDR. The main difference between TER and WER is that the former disposes with the alignment of reference and hypothesis, and instead uses differences in word-counts, see Equation 2.2, where $A(w, d)$ is the number of occurrences of term w in the automatic transcription of document (or story) d, $B(w, d)$ the number of occurrences in the reference transcript, and N the total number of terms in the reference.

$$TER = \frac{\sum_d \sum_w |A(w,d) - B(w,d)|}{N} \times 100\% \tag{2.2}$$

Intrinsic evaluation of transcripts, i.e., evaluation on inherent properties of the transcript, has until now been the standard approach to optimizing ASR performance. But the impact of ASR errors on the performance in a specific scenario of use can only be truly determined if an extrinsic evaluation is done, i.e., using the transcript in its intended context. For SDR, this suggests feeding transcripts into an IR platform and then evaluating the system as a whole. As setting up a collection-specific IR evaluation is rather time-consuming, see Section 2.2.2, and unsuitable for a system optimization workflow, an alternative approach is needed. This thesis investigates the implementation of such an alternative approach to ASR transcript evaluation.

2.1.3 Transcription Alternatives - Lattices

The 1-best literal orthographic transcript that is typical for dictation-type ASR is not the only viable way of transcribing speech. In an SDR context, where the transcript is used as an information source for IR and need not be presented to a user directly, any computer readable representation may be suitable. One potentially interesting alternative representation is a speech recognition lattice. In essence, an ASR system scores multiple possible transcripts for their match with the speech data. Generating the 1-best output is a process where the most likely candidate is selected, given the search space. This search space can be represented using a confusion network or lattice: a graphical structure that represents (part of) the search space that was evaluated by the speech recognition system, typically including the various likelihood scores [Young et al., 2006]. A lattice not only contains the most likely candidate for each position, but also other potential transcripts that were considered during the decoding process.

Figure 2.3 is an example of a very small lattice in which at most three alternative terms for each position are shown. Typically an ASR process considers thousands of candidates for each position, but visualization of such a search space would be counter-productive. This particular example shows 25 numbered nodes, which are connected by arcs which represent words and scores. The scores are transcription likelihoods and depend on the context in which they occur, so a lattice may have the same term occurring multiple times at a single position, but each time with a different associated likelihood, due to a different context. Lattices are a very rich source of information and can be extremely useful for optimization of systems, as they allow for quick modifications of likelihoods and may subsequently produce a different 1-best transcript. Rescoring of lattices is a powerful way of estimating the impact of changes in language model scoring without the need to redo an entire ASR task. One potential drawback of a lattice representation is that it is difficult to determine its intrinsic quality, as the desired properties of a lattice are entirely dependent on its application.

Figure 2.3: *A 25-node lattice with words and scores on arcs.*

The use of lattices in SDR applications has been rather limited so far. One of the main reasons seems to be that a straightforward implementation in which each term in a lattice is included in an index, but with a reduction in weight based on its likelihoods, has not resulted in appreciably improved performance [Saraclar, 2004]. A likely reason for this is that many ASR systems operate with a WER of under 30%. When less than 30% of terms in the 1-best transcript are incorrect, this means that all alternatives are incorrect for 70% of word positions. When the correct word is in fact OOV or the error is the result of severe noise, the correct term is unlikely to be present in the lattice at this position, making the potential for improvement (much) less than the actual error rate. Weighting down the transcription alternatives is not really a solution either as this will also weigh down correct alternatives for errors in the 1-best, reducing potential benefits even further.

Although direct indexation of a lattice may not have yet resulted in improved performance, one could still look for other manners to benefit from their use in SDR. Rescoring of lattices is a relatively quick process when compared to a full ASR run, and may result in an improved 1-best, especially if additional information can be added to the decoding. Improved language models which are based on user-provided information for example may be beneficial without the need for new acoustic models. Another potential application of lattices in SDR is by incorporating transcript alternatives in combination with the use of proximity information [Chelba and Acero, 2005].

2.1.4 Confidence Scores

One of the most immediately useful applications of speech recognition lattices is the ability to calculate confidence scores. Imagine a situation where we want to know not just the most likely transcription of an utterance, but also the probability that this transcript is correct. A standard Viterbi [Viterbi, 1967] decode provides us with a likelihood score, but this is just a very small number that is highly dependent on the duration of the utterance and its other acoustic properties. In order to know the probability that the 1-best output is correct, we need to know the likelihood for every possible transcript, and normalize for the sum of all likelihoods. This is usually not feasible due to the sheer amount of potential transcripts. Lattices however, contain a limited subset of the search space, typically containing only the most likely transcripts. If we then assume all word sequences not supported by the lattice to have a negligible likelihood, we can normalize the utterance-likelihood using only the paths in the lattice. This gives us the probability of a transcript being correct, given a certain search space (the lattice). Such a probability is often referred to as 'posterior probability' or confidence score.

Confidence scores open up new ways of optimizing the performance of an ASR system, as they can not only be calculated for an entire utterance, but also for each word. Calculation of a word-posteriors is done by summing the likelihoods of all paths q passing through a certain link l (representing a term in the transcript), normalized by the probability of the signal $p(X)$, which is the sum of the likelihoods over all paths in the lattice [Evermann and Woodland, 2000]. For example, Equation 2.3 can be used to calculate posteriors, with p_{ac} and p_{lm} being the acoustic and linguistic likelihoods, and γ a scaling factor. Such posteriors are suitable for decoding a lattice into a 1-best transcript [Mangu et al., 2000] and using 'consensus decoding' optimizes for errors at the word level, generally resulting in a lower WER than Viterbi decoding, which optimizes for errors at the utterance level.

$$p(l|X) = \frac{\sum_{Q_l} p_{ac}(X|q)^{\frac{1}{\gamma}} p_{lm}(W)}{p(X)} \tag{2.3}$$

One of the desirable properties of word-posteriors is that all links are assigned a posterior, and as a consequence of the normalization, the sum of the posteriors for all links that cover a certain acoustic frame add up to one. This is especially nice in the context of lattice-based indexing, as the sum of all word-posteriors in a lattice is equal to the expected number of words in the utterance. An important limitation of posteriors is that they only take into account what is explicitly represented in the lattice: if the actual word that was uttered at a particular time was not part of the search space, or not in the lattice due to pruning, then the posteriors of the terms that are in the lattice at this position still add up to one, despite none of them being correct! On average, posteriors are therefore higher than the statistical probability of a term being correct, so one must be careful when using a posterior literally as probability.

2.1.5 Summary

Automatic speech recognition is a statistically motivated process in which the combination of training material and parameter optimization largely determine speech transcript quality. The default configuration of many speech recognition systems is optimized for dictation-type use, targeting a minimal WER for the 1-best transcript. When used in an SDR context, this may not be optimal as not only the amount of errors but also the type and content of the affected words is important. Optimizing ASR systems is typically a matter of correctly setting various parameters, and adapting the lexicon and the acoustic and linguistic models to a specific task, but this can only be done efficiently if we properly evaluate the quality of the transcript in context. In addition, by using lattices and confidence scores, more information from the recognition process can be obtained, which may subsequently be used for enhancing the performance of an SDR system.

2.2 Information Retrieval

For many users Information Retrieval is almost synonymous with web search, as this is the most high-profile use of the technology these days. However, (automatic) IR has been studied long before the advent of the internet. The development of the SMART IR system [Salton, 1971] and the availability of an evaluation method through the Cranfield experiments in the 1960s [Cleverdon and Keen, 1966] gave researchers the tools they needed to make meaningful progress in the field. Several decades later, in 1992, the Text Retrieval Conference (TREC) [Voorhees and Harman, 2005] was organized for the first time. This series of evaluation benchmarks was specifically aimed at IR on large collections, making it possible to see how well the systems that had been developed up until then would perform when scaled up to large-size collections. Since the initial TREC, there have been many developments that triggered the introduction of novel approaches, such as the increasing popularity of the internet and web search.

The second part of this chapter is provides an overview of what we consider 'traditional' information retrieval techniques. The implementation of IR that we use in this thesis does not deal with issues such as interrelatedness of documents or multilingual aspects as these do not relate directly to the research questions we aim to answer. We also do not attempt to maximize IR performance through techniques such as document or query expansion, as these introduce additional parameters and may detract from the core performance we wish to analyze.

After summarizing the basic IR process in Section 2.2.1, Section 2.2.2 provides a short summary of what the Cranfield paradigm entails as this is still the basis of most current IR evaluation. This is followed by Section 2.2.3 on one of the most successful approaches to document ranking in the TREC IR benchmarks: Okapi/$bm25$, which is the ranking method that was used for all our experiments in this thesis. The TREC benchmarks on spoken document re-

trieval are discussed in Section 2.2.4, along with the TDT-2 collection that was used for those benchmarks and that is also the collection that we used to test our methods. Section 2.2.5 gives a short introduction to known-item retrieval, a different approach to information retrieval, which inspired us in developing a query generation algorithm. Finally, we present some overall thoughts on IR in relation to spoken document retrieval in Section 2.2.6.

2.2.1 Implementation

> Information retrieval is finding material (usually documents) of an unstructured nature (usually text) that satisfies an information need from within large collections (usually stored on computers).

This is the definition of information retrieval as given in [Manning et al., 2008], one that is broad enough to include Spoken Document Retrieval as a form of IR. Effectively, an Information Retrieval *system* has the task of doing what is described in the above definition automatically. This is typically implemented by scoring all documents in a collection for their similarity with an information need as expressed through a query, and producing the documents in descending order of similarity.

Making the conversion between fuzzy information need and query autonomously is not possible within the current state-of-the-art of computer science. Such an endeavor would probably require too high a level of artificial intelligence in communicating with the user as well as interpretation of the content of the collection. As formulating a query is an essential task in IR, this makes automatic systems mostly just an aid to a human searcher where the combined efforts of both user and machine produce better results than either could achieve individually. Most applications of information retrieval systems require the user to translate their information need into an unstructured query. Usually great care is taken to make this as easy as possible by making the system more robust towards suboptimal query formulation, for example by using query expansion techniques.

Especially when collections are large, the guidance that an IR system can provide is crucial for efficiently searching the collection. All contemporary IR systems rank the documents in a collection according to their likelihood of being relevant towards a query. As such, the user's task is changed from doing a random or linear search through the collection into a guided one where the first documents are the most likely to satisfy the needs. In unstructured collections, specifically speech collections, even a poor performing speech retrieval system is likely to be of great help as compared to doing the task manually. The task of comparing query terms with collection content can usually be done extremely quickly when compared to the user's task of judging the relevance of a result. This makes it common practice to refine a query based on earlier results to better reflect the information need with respect to the content of the collection and/or the properties of the retrieval system. As a result, information retrieval systems are often extremely useful, even when they are far from perfect.

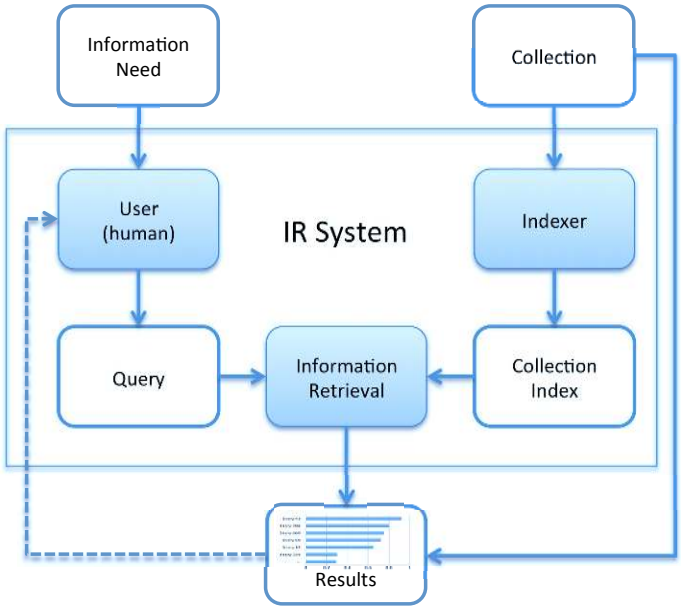

Figure 2.4: *Overview of the information retrieval process. Retrieval results are used as feedback for the task of converting an information need into a query.*

Figure 2.4 provides an overview of the information retrieval process. On the left side, a human user is included as part of the system, as it is the user's task to create a query based on an abstract information need. There is no universally correct way of doing this, and the behavior of a system on a given collection may be cause for refining the original query into something more likely to give acceptable results.

The information retrieval process itself as shown on the right side of Figure 2.4 can be implemented in various ways, but is universally based on superficial similarity between query and retrieval unit (typically called documents or stories). The choice of retrieval unit is what determines the *granularity* of the results. When searching through a library of books, one may be initially satisfied with a list of potentially relevant books, whereas a further search within a book may produce chapters, paragraphs, or page numbers as retrieval results. The granularity of search ideally depends on a user need: in order to quickly get an answer to a specific question, one wants smaller grains than when looking in a general way for information on a subject. In practice, the granularity is often dictated by the collection: searching the internet brings one to specific articles/pages rather than to an entire site, and a search through scientific publications will typically result in links to individual articles. In order for a collection to be automatically searchable, it must be i. in a computer readable representation, i.e., plain text, XML, speech transcription lattices, etc., and ii.

divided into natural retrieval units, i.e., chapters, paragraphs, pages, sentences, utterances, etc.

2.2.2 Cranfield and IR Evaluation

The information retrieval system as described in the previous section includes a human user. This adds some complication to the issue of evaluating the effectiveness of an IR system, as this not only depends on the automatic component, but also on the user who continually refines their queries. It could be argued that since the task of the automatic system is to aid a user in the process, it is the improvement in search effectiveness for the user that should be scrutinized. Doing a user-driven full evaluation leads to huge impracticalities in i. finding enough representative users, ii. costs from paying these users, and iii. that it would lead to huge delays in evaluation which may diminish the possibility to quickly make iterative improvements through parameter tuning. Evaluation of IR systems is therefore done only on the automatic steps, with queries as static information needs, and based on a user-generated ground truth.

Scientific investigation into information retrieval started with the so-called *Cranfield* experiments [Cleverdon and Keen, 1966] in the 1960s by Cyril Cleverdon. The evaluation method that was developed at that time is still the basis for evaluation in all major information retrieval benchmarks such as Text REtreival Conference[5] (TREC), Cross Language Evaluation Initiative[6] (CLEF) and NII Test Collection for IR Systems[7] (NTCIR). The method is based on a simplification of the process and the user needs through three main assumptions [Voorhees, 2002]. The first assumption is that a relevant document can be identified through topical similarity, in other words: the relevance of a document is dependent only on a query and not on the user. The second assumption is that the relevance of each document towards a query can be determined objectively and can be used as a ground truth, and the final assumption that needs to be made is that the ground truth is known and complete. Having these assumptions match reality seems more easily achievable on the collection that Cleverdon investigated than on the typical collections found in the current benchmarks, because of both size and content related issues. However, the history of TREC evaluations has shown that the results that were achieved when these assumptions were adopted for modern benchmark evaluation indicated a relatively stable ranking of systems over a variety of tasks, vindicating their suitability towards evaluations beyond the Cranfield type collections [Harman and Voorhees, 2005].

The basis of Cranfield-type evaluation are the metrics *precision* and *recall*. The former is the proportion of documents in a results list that is relevant towards a query, and the latter equals the proportion of the total number of relevant documents in the collection that is included in the results. Typically there is a trade-off between the two: producing all documents in a collection

[5] http://trec.nist.gov
[6] http://www.clef-initiative.eu
[7] http://ntcir.nii.ac.jp

will result in perfect recall, but probably low precision, whereas producing few (high-quality) results should give high precision, but low recall. Both of these measures are set-based and as such not directly suitable for evaluation of ranked lists.

The measure that is used for the evaluation of ranked lists generated by IR systems is Average Precision (AP). It is the average of the precision of the sets which contain the results up to each relevant document. See Equation 2.4, with M the number of relevant documents in the collection and R_k the collection of all documents from rank 1 until relevant document k. In IR evaluation typically multiple queries (around 50) are used, for which the Mean Average Precision (MAP) is calculated by taking the mean of average precision for each individual query.

$$AveragePrecision = \frac{1}{M} \sum_{k-1}^{M} Precision(R_k) \tag{2.4}$$

The main limitation in the use of MAP is that it is notoriously hard to determine for an ad-hoc collection. The Cranfield assumptions indicate that one needs to define the relevance of each document in a collection as a function of each query in the evaluation. Even for modestly sized collections this is quite a big task, but for the type of collections that are typically the subject of IR research it is nigh-on impossible, making it necessary to optimize the effectiveness of time spent on the judgment process. This is done by making judgments a binary decision on a document's relevance towards a query, making no distinction between levels of relevance, and by judging only candidates that were selected from a pool of results of preliminary retrieval experiments. Such deviations from the original Cranfield experimental configuration have been shown to have a relatively limited impact on system ranking [Voorhees, 2000, Voorhees, 2002], but cannot change the fact that a lot of time has to be invested to obtain a sufficient number of relevance judgments.

It is important to observe that MAP cannot be safely interpreted as an absolute performance level. The same system is likely to give very different MAP depending on the collection and the set of queries that it is applied to. As such, MAP is mainly useful as a tool for comparing various configurations for optimizing towards a certain collection and/or information need, or in a benchmarking context. Alternatives for MAP have been developed, e.g. [Büttcher et al., 2010], but all of the experiments in this thesis use a traditional approach to IR and use MAP as a baseline evaluation approach. Section 2.2.5 briefly discusses an alternative approach to IR and its standard evaluation method, as it pops up briefly in Chapter 5.

2.2.3 Term Weighting and Okapi/$bm25$

Most algorithms for determining the relevance of a document for a given query are based on the calculation of weights. The higher the weight that is assigned to a document, the more likely it is to be relevant. In practice, most systems

assume a bag-of-words approach to queries, meaning that each query term is individually matched with documents and its contribution to the total relevance is independent of other query terms.

One of the earliest approaches to weighting documents towards a query is $tf_{t,d} \times idf_t$ which stands for the *term frequency*, i.e., the (normalized) number of occurrences of a specific query term in a particular document, multiplied with the *inverse document frequency*, i.e., (the logarithm of) the inverse of the proportion of documents in the collection that contain the term. See Equation 2.5, where $c_{t,d}$ is the count of term t in document d, c_d is the total term count in document d, N is the number of documents in the collection, and c_t is the number of documents containing term t. One can calculate a $tf_{t,d} \times idf_t$ value for each query term and document and combine these into a final score per document using, for example, a vector space approach [Salton et al., 1975].

$$tf_{t,d} \times idf_t = \frac{c_{t,d}}{c_d} \times log\frac{N}{c_t} \tag{2.5}$$

The family of weighting functions that was introduced with the Okapi system [Spärck-Jones et al., 2000] is somewhat similar to $tf_{t,d} \times idf_t$ in that it has components that are similar to tf and idf, but differs in the way these are calculated. The $bm25$ idf-component is defined as Equation 2.6, and the tf-component as Equation 2.7, where c_D is the average document length over the entire collection, and k_1 and b are two tuning variables.

$$log\frac{N - c_t + 0.5}{c_t + 0.5} \tag{2.6}$$

$$\frac{c_{t,d} \times (k_1 + 1)}{c_{t,d} + k_1 \times (1 - b + b \times \frac{c_d}{c_D})} \tag{2.7}$$

In the standard $tf_{t,d} \times idf_t$ weighting, each occurrence of a query term in a document causes a linear increase in the weight, i.e., a document containing a query term twice has double the weight of a document containing it once. This was found to be counter-intuitive as additional occurrences of the same term do not necessarily make a document more relevant in a linear way. The tuning parameter k_1 is introduced in order to reflect the expected non-linearity of the contribution of additional term occurrences to the expected relevance. A lower value of k_1 implies a reduced linearity in the relationship: additional occurrences of the same query term have an increasingly small effect on the weight. The tuning variable b works in a similar way for document length, where a value of b that is less than one implies a decreased linearity.

Combining Equation 2.7 and 2.6 and adding a query-term normalization into the $bm25$ function results in Equation 2.8, where Q is the number of terms in the query, and c_q the count of term q in the query. The first part of this equation is a non-linear interpretation of duplicate terms in the query as multiple occurrences of the same term in a query are not necessarily indicative of a linearly increased importance of that term for the relevance of documents. In our experiments we

did not use this additional normalization as our queries contained no duplicate terms.

$$bm25 = \sum_{q=1}^{Q} \frac{(k_3+1)\times c_q}{k_3+c_q} \times \frac{c_{t_q,d}\times(k_1+1)}{c_{t_q,d}+k_1\times(1-b+b\times\frac{c_d}{c_D})} \times log\frac{N-c_{t_q}+0.5}{c_{t_q}+0.5} \tag{2.8}$$

The Okapi/$bm25$ approach to term-weighting has been used extensively in the various IR benchmarks and has been shown to provide good results [Spärck-Jones et al., 2000]. It has parameters whose function is relatively easy to understand, making this ranking function especially useful for experimentation purposes.

2.2.4 TREC SDR and the TDT Collections

The Text REtrieval Conference[8] (TREC) is an annual event that is designed to encourage research on text retrieval for realistic applications by providing large test collections, uniform scoring procedures, and a forum for organizations interested in comparing results [Voorhees and Harman, 2000b]. As part of this event, a multitude of evaluation tasks is performed by many labs in the IR field from all over the world. These tasks are called tracks and although they typically deal with textual resources, from TREC-6 in 1997 until TREC-9 in 2000 a Spoken Document Retrieval track was included.

The best systems in these TREC SDR tracks uniformly used an automatic speech recognition system in a dictation type configuration to produce a 1-best transcript of the speech. A 'standard' information retrieval system was used to perform the ranking of documents. As not all participants would have access to a state-of-the-art speech recognition system, a baseline automatic transcript was provided. This way participation would be open to any interested labs, including those that were primarily interested in retrieval research.

For TREC-6 [Garofolo et al., 1997] a 50 hour collection of broadcast news stories was used with 47 test topics, with the goal to find a single target story – a known item retrieval task. The granularity or size of the retrieval units was much smaller than for typical textual retrieval tasks with an average of 276 words per story over a total of 1451 stories. Word error rates of the various transcripts ranged between 35 and 40%. The best system retrieved the target story as the first result for 37 out of 47 topics on a human made reference transcript, and 36 times on a speech recognition transcript, indicating a very small impact from speech recognition errors on this task for the top system.

These results were not particularly insightful as the collection was much smaller than typical textual collections and the task was relatively easy. For the TREC-7 SDR track [Garofolo et al., 1998] it was therefore decided to use a more typical ad-hoc retrieval task instead of the known-item task, but the collection used was still small at 87 hours and 2866 stories, with an average of 269 words

[8] http://trec.nist.gov

each, but a median story length of only 94 words. There were 23 topics with an average of 17 relevant stories per topic. Word Error Rates of ASR transcripts of this collection were typically between 25 and 35%. The best system achieved a MAP of .5668 on the reference transcript and .5075 on a speech recognition transcript. As with TREC-6 it seemed as if the impact of speech recognition errors on retrieval performance was relatively mild (a little over 10% reduction in MAP) for the given task, but this collection was also considered too small to make any definitive judgements.

The TREC-8 SDR track [Garofolo et al., 2000b] adopted the evaluation method of TREC-7 SDR, but used a more realistically sized collection: TDT-2 from the Topic Detection and Tracking benchmark [Fiscus and Doddington, 2002]. This collection contained 21754 stories spanning 557 hours of audio from the Broadcast News domain. 49 topics were developed with a total of ∼1800 relevant documents. Automatic speech recognition resulted in word error rates ranging between 20 and 30%, whereas for this collection the references were mainly closed captions and radio transcripts with a 7-15% error rate. The best system achieved a MAP of .5598 on the reference transcripts and .5539 on the automatic transcripts. The impact of speech recognition errors was thus reduced to around 1%, resulting in the SDR problem being famously declared solved for this type of collection and task. The last TREC-SDR [Garofolo et al., 2000a] was held at TREC9 and was very similar to TREC-8, using the same corpus with a different set of 50 topics. Results for the best system were .5268 for reference and .5194 for ASR transcripts, a further indication that SDR using this type of collection and this type of task could be considered a solved problem.

The TDT Speech Collections One of the most interesting things to come out of TREC SDR, apart from the somewhat surprising finding that transcript noise (errors) have little impact on retrieval performance, is the availability of a test bed and methodology that can be used for many types of evaluations. In addition to the TDT-2 collection that was used for TREC-8 and TREC-9 SDR, there is also the TDT-3 collection which is similar but uses more recent broadcasts and contains some amount of Mandarin Chinese speech.

The size of the TDT-2 [Cieri et al., 1999] and TDT-3 [Graff et al., 1999] collections seems to enable true information retrieval as opposed to something akin to word spotting that would occur on smaller collections. In addition, there is not only a human-generated reference transcript available, but several of the labs that participated in the TREC SDR benchmarks have released their own ASR transcript, making for at least seven different automatic transcripts of the TDT-2 collection being available for research purposes.

Another potentially interesting subject for study in the context of SDR is the role of story boundaries. The TDT-2 collection has been used to study the effectiveness of automatic story segmentation in the context of the Topic Detection and Tracking benchmarks [Fiscus and Doddington, 2002], and both human and automatically generated segmentations are available. The total number of IR evaluation queries available for the TDT-2 collection is 99 (49 from TREC-

8 and 50 from TREC-9), with a total of ∼4000 relevant documents. For the TDT-3 collection there are two sets of 60 queries with ∼12000 relevant documents. As such, the TDT-2 and TDT-3 collections provide ample opportunity for assessing strategies for improving SDR performance. The main downside of these collections is that they are relatively easy for an ASR system to transcribe as the speech type is almost exclusively read speech. The same criticism seems to be true for the IR part, as MAP scores of more than .5 are rarely seen in the textual tracks of TREC and indicate an especially easy task. The high relative performance obtained using automatic transcripts also means that there is little room for improvement on the results that were obtained in 1999 and 2000 on the TDT-2-based SDR tasks. So for the development of novel IR or SDR approaches with the aim of improving MAP, this collection provides little research opportunity. For our task of investigating the feasibility of a novel evaluation approach for ASR transcripts in an SDR context, the collection has many unique properties whose presence is instrumental in helping us answer our research questions.

2.2.5 Known Item Retrieval and Mean Reciprocal Rank

Outside of the Cranfield framework, a slightly different task for which evaluations have been set up is known-item retrieval. The main contrast with standard IR is the explicit assumption that the user is expected to be satisfied with a single document and has some knowledge of what such a document may contain, but does not necessarily have enough knowledge of the collection to find it directly. There are several variations on this task, for example, finding out what the latest movie of a given director is, instructions to the use of a certain appliance, or general question answering scenarios [Voorhees and Tice, 2000].

Concepts such as precision or recall are less pertinent, as the users are not expected to look at any documents beyond the first satisfactory item. Evaluation in a known item retrieval scenario is therefore typically not done with MAP, but uses Mean Reciprocal Rank (MRR) [Voorhees, 1999]. The MRR is calculated using equation 2.9, with N the number of queries considered and $r1$ the rank of the first (typically only) relevant result.

$$MRR = \frac{1}{N}\sum_{n=1}^{N}\frac{1}{r1_n} \tag{2.9}$$

2.2.6 Summary

Automatic information retrieval systems attempt to solve only part of the problem of matching an information need to a document, as a human is an essential part of the process. Evaluation of the automatic part of the IR process is complicated, as establishing a ground truth reference requires a large investment of time. Some of the techniques that have been successfully used for reducing the required number of judgments on textual judgements, such as pooling the

output of several IR systems to make a pre-selection of interesting content, are potentially less effective for speech content, as transcript errors could penalize certain documents and speech types uniformly for all systems. Obtaining relevance judgments for a speech collection for which no manual reference transcript is available is therefore extremely difficult, and likely to be practically unfeasible.

The results of the SDR experiments in the context of TREC benchmarks have shown that speech recognition is not necessarily a bottleneck in the performance of SDR systems for broadcast news. Evaluations have however only considered the overall quality of the results and not investigated if there may have been transcript-noise-induced bias in the retrieval results. There is also still much unknown about the behavior and usability of SDR systems on other types of collections. Speech retrieval on cultural heritage for example may pose additional challenges for which novel solutions need to be developed [van der Werff et al., 2007]. We probably need more variety in the types of queries and speech collections that are available for research to get a better idea of where the true challenges lie for this type of IR application.

2.3 Conclusion

Automatic speech recognition and information retrieval are both relatively mature fields of research with well-established methods for technical implementation and evaluational approaches. The results of the TREC SDR benchmarks have shown that for certain types of collection, ASR performance is 'good enough' and little can be gained from an SDR point of view by further reduction of error rates.

Building an SDR system for an arbitrary collection, for example from the cultural heritage domain, has so far proven to be much more problematic than the results from the TREC SDR benchmarks suggest [Oard et al., 2006]. Word error rates are typically much higher for non-broadcast news speech and evaluation or tuning of the IR system is very difficult as relevance judgments for ad-hoc queries and collections are usually virtually impossible to create. In general, many of the lessons learned from experimentation in benchmarking conditions cannot be applied to an environment that is inherently resource-poor.

ASR systems are typically optimized for delivering a 1-best transcript with the least amount of errors, but information retrieval system may be better served by an output that is less deterministic and provides the likelihood of the presence of a terms. Similarly, IR systems, including those used in the TREC SDR benchmarks, typically treat the collection as a reliable source of information, thereby discounting the presence of errors in the transcript.

In order to improve SDR performance, it is not enough to simply optimize ASR and IR in isolation, but one needs to look into ways to adapt the systems so as to play towards their strengths. Optimizing ASR output for intrinsic qualities (the lowest WER) may not result in an optimal system configuration in the context of SDR. When implementing an SDR system with known compro-

mises in performance due to transcript noise or accessibility of source material, one cannot assume a one-size-fits-all approach and must take care to optimize each component for the particular demands of the expected usage. Extrinsic evaluation of ASR transcripts can provide better guidance as to the demands of the application and enable the creation of an integrated system for SDR, rather than a simple concatenation of solutions from separate fields of research.

3

Automatic Story Segmentation

The task of a spoken document retrieval system is to rank fragments in a speech collection for their relevance toward an information need expressed as a query. Two requirements must be met for a collection to be searchable: i. it must be in a computer-readable representation, and ii. it must be divided into natural retrieval units. In the case of SDR, the former is typically solved using an automatic speech recognition system which converts audio into text, but the latter requires some additional effort as speech recognition transcripts lack a suitable division into retrieval units. The typical textual clues for content-based (story) breaks, such as chapters, sections, and paragraphs are not found in basic speech transcripts. Given the fact that many of the collections that are potential targets for SDR can be hundreds or even thousands of hours long, this means there is a need for an automatic approach to segmenting speech transcripts into coherent stories.

In the TREC[1] IR benchmarks – which have proven highly influential when it comes to evaluation of IR systems – the search task is carried out on a large collection of individual stories. A story is defined as a cohesive segment (mostly of news) that includes two or more declarative clauses about a single event. A topic is a user need statement which is a more elaborately worded version of a query. For example topic number 76 from the TREC-8 SDR track: 'List documents pertaining to the bombing of the World Trade Center in New York and to the trial and conviction of Ramzi Ahmed Yousef and other alleged conspirators', can be translated into the query: 'bombing World Trade Center New York trial conviction Ramzi Ahmed Yousef conspirators'. The basic task is to match topic descriptions (or queries) to individual stories. One of the fundamental assumptions in traditional TREC-style IR is that all textual content can be approached as a collection of stories and that a user need can be served by a subset of that collection, and be defined by a topic or query. Because TREC-style IR is such a well researched subject, SDR has also been approached in this manner [Garofolo et al., 2000b].

What constitutes a natural retrieval unit depends on the collection and the information need. For some collections it may be reasonably easy to identify 'stories', e.g., Broadcast News (BN) may be divided into separate news events

[1] http://trec.nist.gov

rather straightforwardly, but for 1+ hour radio shows this could be more difficult. Spoken content, especially spontaneous speech, due to its very nature can be quite unstructured, making it difficult to determine where a conversational topic begins and ends. Besides this inherent lack of structure, automatic speech transcripts do not contain any of the structural cues one would expect in textual content, such as chapters, paragraphs or sentences. Since most current solutions for automatic speech recognition do not explicitly address the issue of story boundaries, it must be dealt with in a post-processing step. All serious attempts at automatic story segmentation so far can be classified under two distinct categories [Manning, 1998]: statistical information extraction and exploiting lexical cohesion. Sections 3.1.1 and 3.1.2 provide more information on approaches in each of these categories and how they may be implemented.

Within the context of the TDT benchmarks [Wayne, 2000], the most successful approaches to automatic segmentation of speech transcripts have used statistically motivated methods. Systems were trained on large amounts of manually segmented speech from the same domain as the target collection. It is unclear how well segmentation models that were trained on BN perform on non-BN speech transcripts, but given the typical dependence on certain indicator-phrases, such as 'this is CNN,' or 'signing off,' they are unlikely to be particularly suitable for collections from most other sources and domains. For spoken content, many of the collections that are manually segmented are from the BN domain. However, many collections that are seen as candidate content for SDR are rather dissimilar to BN, e.g., interview collections, non-news radio broadcasts, and historical audio collections.

Our research interest in this chapter is in the feasibility of automatic segmentation of speech transcripts for the purpose of SDR. We focus on lexical cohesion-based techniques as such methods are expected to be more widely applicable in non-BN scenarios. Five algorithms are compared: three established approaches (duration-based segmentation, TextTiling, C99) and two novel methods (WordNet-based and Query-specific Dynamic Segmentation Algorithm). An important difference with earlier research into this subject is that performance was evaluated both intrinsically using a segmentation cost-function, i.e., by comparing the absolute positions of the generated boundaries with a human-made ground truth, and extrinsically using MAP, i.e., through a full SDR evaluation, comparing retrieval performance using indexes built on the automatic and the ground truth boundaries. The research questions that we intend to answer are:

- Does extrinsic rather than intrinsic evaluation of artificial story boundaries lead to different conclusions with regards to the best choice of segmentation method?

- Which is the best method for automatic story segmentation without using additional resources in the context of SDR, based on extrinsic evaluation?

- What is the impact of artificial story boundaries on retrieval performance of an SDR system when compared to a reference segmentation?

The rest of this Chapter is organized as follows: Section 3.1 provides a brief overview of earlier approaches to segmentation, which is followed by a description of the segmentation methods we used in our experiments in Section 3.2. Sections 3.3 and 3.4 describe the design of our experiments and their results. A conclusion and discussion is provided in Section 3.5. The work in this chapter was published earlier in [van der Werff, 2010].

3.1 Previous Work

Automatic segmentation of textual data into topically cohesive units is used for several applications, for example automatic summarization [Barzilay and Elhadad, 1997], and information retrieval [Dharanipragada et al., 1999]. Story segmentation for spoken document retrieval can be defined as dividing an audio stream into coherent segments which are suitable for retrieval purposes. In the case of a broadcast news speech corpus for example, stories are typically complete and cohesive news reports on a particular topic. As this is a natural segmentation that is inherently present in such broadcasts, such a task is well-defined; BN-type stories have topic shifts that are relatively coarse-grained, i.e., one can go from a nuclear disaster to a financial crisis from one story to the next. When dealing with collections that contain spontaneous speech or interviews it may be much more difficult to determine what 'natural boundaries' are. Often, the best segmentation depends on the information need and on the nature of the collection. Whether a segmentation is 'correct' is therefore task dependent, but usually evaluation is done intrinsically or task-independent, i.e., by comparing automatically generated boundaries to manually defined boundaries, without taking the expected use into consideration. Most of the approaches that have been developed and studied in the past have focused on a well-defined task for which stories were relatively easy to define, and the aim of the system was to find these natural segments as accurately as possible.

3.1.1 Statistical Approaches in TDT

The Topic Detection and Tracking[2] (TDT) program which was part of the DARPA Translingual Information Detection, Extraction, and Summarization (TIDES) project, started in 1997 and had open evaluations every year until 2004. Within this program five tasks were defined: Story Segmentation, Topic Tracking, Topic Detection, First Story Detection, and Link Detection. The first of these, story segmentation, was investigated in 1998 on the TDT-2 collection [Cieri et al., 1999] and 1999 on the TDT-3 collection [Graff et al., 1999].

Several labs created systems for automatic story segmentation. The MITRE Corporation [Greiff et al.,] built a Bayes classifier using three textual cues: overlap, start-trigger, and end-trigger, and two audio cues: low energy and change of energy. Using a collection of manually segmented and transcribed speech, each of the five cues could be associated with the likelihood of a story

[2] http://www.itl.nist.gov/iad/mig/tests/tdt/

boundary occurring. For segmentation the sum of log-probabilities of the five cues was taken and compared to a threshold to decide on the presence of a boundary. Training was done on a source-dependent basis, i.e., material from 'Voice Of America' resulted in different segmentation models than material from the 'American Broadcasting Company'.

IBM created a system that used a combination of decision tree and maximum entropy models [Franz et al., 1999], with lexical, prosodic, semantic and structural features as input. The input features for the decision tree were the duration of non-speech events in the ASR output, the presence of key words and bigrams, and the distribution of nouns on either side of a proposed boundary. For the maximum entropy model these were padded with n-gram-based triggers, and large-scale properties of broadcasts such as time slots for commercials. The features were trained on a source-dependent basis.

CMU used a system that was based on a maximum entropy framework with a log-linear model [Carbonell et al., 1999]. Both 'topicality' and 'cue-word' features were used, the former were constructed based on a minimum divergence criterion with respect to a smoothed trigram default model, the latter were learned on a source specific basis. Features were learned from all fields in the speech transcript files, including speaker cluster id, silence duration, and document source. For each of the sources around 200 features were induced.

All of these statistical approaches used manually segmented speech as training material to create models. Some features were so collection-specific that using them on broadcast news material from a different source/network than they were trained on, would compromise performance. Although these approaches may have been suitable for the demands of the TDT-2 and TDT-3 segmentation benchmark task, they are rather impractical for use on non-BN or 'surprise' data, due to the typical sparsity of training material for such collections.

3.1.2 Lexical Cohesion-based Approaches

The assumption underlying lexical cohesion-based approaches to automatic story segmentation is that cohesion between words within a story is higher than between words from separate stories. By explicitly identifying cohesion, one can identify places within a text that show relatively little cohesion, indicating a likely boundary between stories. Cohesion can be identified in several ways, e.g., repetition, synonymy, and specialization/generalization relations between words.

The TextTiling [Hearst, 1997] approach to segmenting text attempts to identify passages or subtopics using paragraphs as its basic units. There is no use of any kind of discourse cues, only a strict reliance on patterns of lexical co-occurrence and distribution. This approach is probably best suited to coarse-grained topic shifts, as it is based on the assumption that (sub)topic change can be detected through changes in vocabulary. Implementation is by a comparison of blocks of equally-sized token sequences around paragraph breaks using a sliding window approach. This results in a lexical score, which is a normalized inner product of term frequencies in both blocks, indicating the amount of rep-

etition in (tokenized) terms. Positions where repetition is low in comparison to the surrounding blocks are candidate segmentation points. Actual segmentation is done by smoothing the lexical scores for the candidate segmentation points and choosing the most likely positions. Performance was shown to be highly dependent on the material it was used on, with the worst results reported on text that was 'chatty' in nature.

The C99 [Choi, 2000] algorithm uses a cosine measure to determine the similarity between all (stopped, stemmed, and tokenized) sentence pairs in a document. These scores can be entered into a matrix with the sentence numbers along each of the axes. An 11x11 rank mask is then used on this matrix of similarity values to enhance contrast. Segmentation of the text must take place on the diagonal of the matrix, after which each segment will itself form a smaller matrix. For any such matrix a 'density' can be computed, representing the ratio between the sum of similarity ranks and area. Each segmentation will cause the sum of these densities over all segments (called the 'inside density') to increase. The best segmentation position at any given point in the process is the one that results in the highest increase in inside density. Segmentation stops when the increase in inside density drops below a specified threshold. Evaluation was done on (fragments of) textual documents which were artificially concatenated. Performance comparisons between several types of 'random' segmentation and TextTiling showed little difference in error rates in this study, whereas the C99 algorithm gave significantly better performance on this task.

The SeLeCT [Stokes et al., 2004] system uses an approach where text is tagged and morphologically analyzed and normalized. Then WordNet [Fellbaum, 1998], a large lexical database of interlinked sets of cognitive synonyms, and 'a set of statistical word associations' is used to find relationships between tokens (nouns, proper nouns, compound nouns, and nominalized adjectives), which form the basis of the lexical chains. Tokens can form chains based on cohesive relationships: repetition, synonymy, statistical associations, generalization/specialization, and part-whole/whole-part relationships. Boundaries are hypothesized at points which have a relatively large number of chain beginnings and endings. The system was tested using paragraphs as the basic unit for a textual collection and speaker change for speech transcripts. Experiments on the TDT-1 [Allan et al., 1998] corpus showed that the system performed better than TextTiling on the speech transcripts, but worse for the textual documents. By analyzing properties of the collections, it was possible to obtain improved performance of TextTiling, C99, and SeLeCT through better filtering of verbs and identification of referential and conjunctive constructs on the speech transcripts of one source. It was concluded that analysis using only patterns of repetition resulted in the best performance, onto which lexicographic and statistical relationships between tokens could not improve.

3.1.3 Alternative Approaches to Segmentation for IR

For the most commonly used approaches to IR, explicit story boundaries are required for efficient indexing and retrieval. However, efficiency (i.e., speed and

storage requirements) may not always be a top priority. Many spoken document collections are relatively small when compared to textual collections and may only attract a relatively small number of simultaneous users. In those cases, one may be satisfied with a slower system that requires more data-storage, but potentially produces higher quality results.

An alternative to the traditional IR approaches that were mentioned in the previous sections is the use of Hidden Markov Models [Mittendorf and Schuble, 1994] for retrieval. For this approach, boundaries can be generated on-the-fly. The ranking method had some similarities with language model-based IR schemes [Hiemstra, 2001], but passages are automatically isolated based on their match with the query and their contrast with surrounding terms. The entire collection is assumed to be generated by a three-state HMM, with each state represented by a language model (LM): an LM of non-relevant (or general) text, an LM which is based on the query, and finally another LM of non-relevant text. For this HMM an overall likelihood of the collection is calculated, with the alignment of states and text that provides the largest contribution to the overall likelihood is used to segment the collection into three parts. The part corresponding to the query-LM can then be isolated as a retrieval unit. This process can be repeated on the remaining parts until the required number of results is reached.

Another alternative approach to IR is found in a method which selects passages from large documents based on queries and was described in [Salton et al., 1993]. It is based on the idea of selecting increasingly smaller sections of the collection based on some similarity to a query. A text environment/section is considered relevant and presented to the end-user, only if there is sufficient local similarity to the query text.

3.2 Story Segmentation for SDR

The segmentation algorithms that were mentioned in the previous section were all optimized for the task they were tested on. The statistical approaches were used in the TDT benchmarks because of their ability to capitalize on the predictability of the collections, the availability of suitable training material, and their lack of reliance on long(ish) stories. The lexical cohesion-based approaches on the other hand do not require any collection-specific training material, and seem to work best for longer stories. They were typically implemented with somewhat large basic units, i.e., sentences or paragraphs. In all cases, the automatic segmentation systems were optimized for segmenting a textual stream into shorter segments in a manner that was most similar to what humans would do intuitively. The human-made segmentation was used as ground-truth and systems were evaluated based on their ability to produce boundaries at the same (or close) positions, typically using a cost-function.

As our interest is in spoken document retrieval, an intrinsic evaluation as was done in the various approaches mentioned in Section 3.1 may not best represent our requirements of the segmentation. Instead we compare six automatic

segmentation methods for their performance in the context of the calculation of relevance of speech fragments towards TREC-style queries.

For the TREC9 SDR [Voorhees and Harman, 2000a] benchmark, the primary task was for spoken document retrieval in a story boundaries unknown (SU) condition. The task was similar to the one from TREC8 SDR [Garofolo et al., 2000b], i.e, a standard TREC-style IR task on speech transcripts, but this time without using the provided manual story boundaries. For efficient evaluation, the participants were required to produce a ranked list of *positions* into the collection, which could subsequently be mapped onto the 'known' stories and be scored in the usual manner. This made for a somewhat unusual task, in that the list of starting points for a user into the collection (as would be typical for normal textual retrieval, and the TREC8 SDR task), was replaced with a list of positions that were located somewhere within a coherent section with a user needing to find a proper starting point on their own account. As long as the provided positions were anywhere within a relevant section, the results were counted as correct. Although such a scenario is probably not ideal from a user perspective, it does make evaluation rather straightforward. For this reason, we conduct our experiments in a similar manner.

The algorithms we are interested in are: 1. the statistical approach as implemented by IBM [Franz et al., 1999], 2. a duration-based method that was used by the most successful labs in TREC-SDR, 3. TextTiling, 4. C99, 5. a WordNet similarity-score-based segmentation approach of our own design, and 6. a dynamic segmentation method also of our own design. Methods 5 and 6 are novel approaches, at least for this task. For method 1 we use a ready-made segmentation as provided by IBM, this means that there are no alternative configurations that can be investigated for this method. The remainder of this section contains descriptions of the existing methods 2-4, and descriptions of the novel approaches of methods 5 and 6.

To get a better idea of what such segmentations look like, Figure 3.1 contains a visual interpretation of the results of the various segmentation methods. The horizontal axis represents a time-line into a half-hour segment from the TDT-2 collection (specifically the one called 19980201_1130_1200_CNN_HDL). The horizontal blue lines are the segments as they result from the use of the segmentation algorithms we investigate in our experiments. The segments are plotted in a staggered manner for improved visual clarity only. The five methods that are investigated in this chapter are shown, plus the reference segmentation and segments that resulted from a statistically motivated approach by IBM, which is included for comparison purposes.

3.2.1 Duration-based segmentation

The rather straightforward duration-based segmentation method was the most popular with labs that participated in the TREC9 SU SDR task. Of the labs that created their own boundaries, Limsi [Gauvain et al., 2000] and Cambridge University [Johnson et al., 2000] were the most successful and they used this approach.

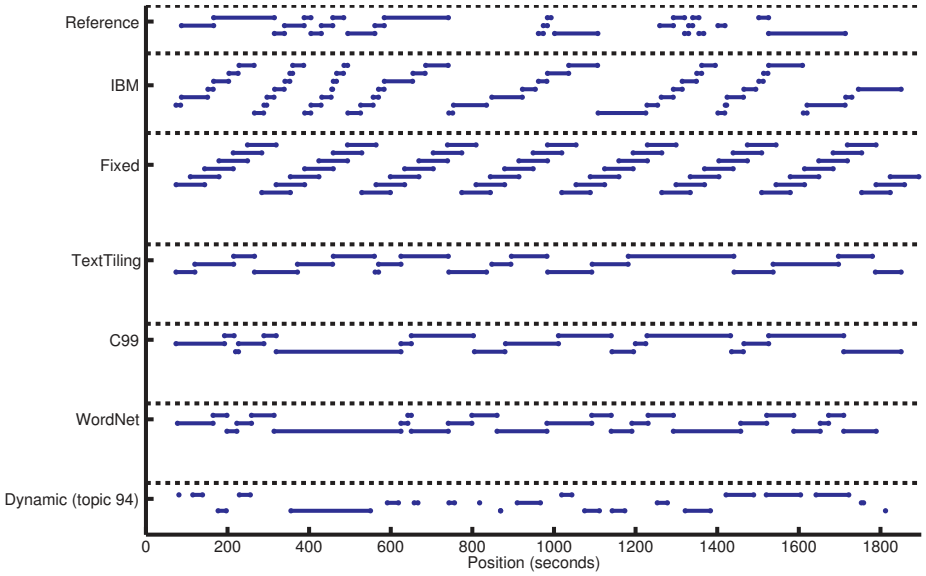

Figure 3.1: *Story boundaries as generated on the document 19980201_1130_1200_CNN_HDL using the methods that are investigated in our experiments. Staggered plotting is only for visual clarity.*

For fixed-duration segmentation, speech or a speech transcript is segmented into sections based only on their duration (in seconds). A window of a predetermined size is overlaid on the transcript and its contents are interpreted as a segment. Windows can be either adjacent or overlapping, resulting in the possibility of one term being part of multiple segments. If overlapping segments are indexed in the traditional manner, this may lead to duplicate results in an IR task. It is therefore important to remove such duplicates in cases where overlapping windows were used for story segmentation. In our experiments we do this by discarding all results that have any overlap with positions in the collection that are also covered by a higher ranking result.

One of the most attractive properties of duration-based segmentation is that it can be applied to speech transcripts without any prior information on language, speech type, or structure, such as speaker changes or utterance boundaries. The only variables to consider are the duration of the segments and the amount of overlap of subsequent segments. The optimal setting for segment duration may be quite dependent on the collection. The TDT-2 collection we use in our experiments contains mainly news items which are relatively short, but interview collections for example, may require longer segments for best IR performance.

3.2.2 TextTiling and C99

The TextTiling and C99 algorithms were briefly explained in Section 3.1.2. We use an implementation from the MorphAdorner[3] Java libraries for our experiments. When using TextTiling, the likelihood of a story boundary is calculated for a number of potential candidates that are defined by initial basic units. In the original TextTiling implementation, paragraphs were chosen as basic units, as in textual documents it was assumed that story boundaries coincide with paragraph boundaries. For C99, sentences were originally used as basic units. Neither paragraph nor sentence boundaries can typically be found in automatic speech transcripts.

To use TextTiling and C99 on speech transcripts we must provide candidate boundary positions, but because they are not included in ASR transcripts, we cannot use paragraphs or sentence boundaries. There are however two other basic structural clues that *are* provided by most ASR systems: utterance boundaries and speaker changes, but there is no guarantee these coincide with natural story boundaries. Typically, utterance boundaries are hypothesized at positions with low energy, and speaker changes are detected from more complex properties of the speech frames. Speaker changes may happen at any time during a broadcast, and may be indicative of a story boundary when the program is hosted by two people who take turns in presenting news items, or for interview collections. On the other hand, it is just as likely that one presenter reads several stories in succession, which would result in speaker changes being a relatively poor basic unit. We therefore decided to focus our efforts exclusively on using utterances as initial basic segmentation units for these two methods.

A potential concern for the use of these methods in an SDR context with the TDT-2 collection is that the reference stories in this collection were relatively short at an average of 173 words, and with half the stories less than 86 words long. Stories in textual collections that are often used in TREC benchmarks are typically much longer by a factor of 5 or more. It is expected that longer stories contain more repetitions which is the basis for segmentation by these two methods. The abundant presence of short stories most likely works against the effectiveness of the algorithms, so we expect these methods to underperform on our collection.

3.2.3 WordNet-based Segmentation

WordNet [Fellbaum, 1998] is a lexical database of English, but equivalents are available in many other languages including most official European languages. In WordNet, nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms, each expressing a distinct concept. These sets are then hierarchically linked to express semantic and lexical relations. The method that is introduced here uses these hierarchical relationships to generate links between basic units. Positions in the collection where the similarities between words on either side

[3] `http://morphadorner.northwestern.edu`

are stronger than for the combination of the words on both sides are potential segmentation points.

Due to the hierarchical nature of the WordNet database, it is possible to express the distance and relative position of two terms in it as a similarity score. Out of many methods that have been proposed for this, we choose the one described by Jiang and Conrath [Jiang and Conrath, 1997] as it uses a combination of distance and information content similarity. We use the Perl CPAN module 'WordNet-Similarity' for calculating the similarity scores which ranges from 0 (for no similarity at all) to 1 (for synonyms).

The edge-based (distance) similarity is the sum of the *is-a* edges in the hierarchy between the two words that are compared. For the node-based (information content (IC)) similarity, IC-based weights are used. The concept of IC in WordNet nodes is inversely related to the probability of encountering an instance of such a node. The further down in the hierarchy one goes, the higher its IC as the concept becomes more specific. For example, take a small section of the WordNet hierarchy which has the node 'vehicle' at the top, with both 'car' and 'bike' as child nodes. The 'car' node subsequently has 'sedan' and 'hatchback' as child nodes. The lowest information content is found in 'vehicle' and the highest in 'sedan' and 'hatchback'. The node-based similarity score between two words is defined as the IC score on the lowest node that contains both terms, in our example: 'sedan' and 'hatchback' have a node-based similarity score equal to the IC score of 'car', whereas 'hatchback' and 'bike' have a score equal to the IC score of 'vehicle'.

We test whether the similarity measures can be interpreted as a likelihood of two terms belonging to the same story. If we assume the likelihood of two terms being part of the same story is equal to the WordNet-based similarity score, then Equation 3.1 describes the likelihood of any set of terms belonging to a single story. M is the total number of terms in the set, and $\widehat{p}_{sim}(l, m)$ is the WordNet-based similarity score between the terms at positions l and m from this set.

$$\widehat{p}_{story}(1, .., M) = \prod_{l=1}^{M-1} \prod_{m=l+1}^{M} \widehat{p}_{sim}(l, m) \tag{3.1}$$

We use $\widehat{p}_{story}$ to calculate, for any potential boundary position, the likelihood of it being a 'true' story boundary. This is determined by the likelihood of all terms in its left context (the extent of which is determined by a variable n) being part of a single story, and its right context being part of a single story, and the likelihood that left and right-side context together *do not* form a single story, see Equation 3.2. $\widehat{p}_{!story}$ is in this case calculated in a similar manner to $\widehat{p}_{story}$, except that instead of $\widehat{p}_{sim}$, we use $1 - \widehat{p}_{sim}$.

$$\begin{aligned} \widehat{p}_{bound}(t) = \quad & \widehat{p}_{story}(t - n + 1, .., t) \times \widehat{p}_{story}(t + 1, .., t + n) \\ & \times \widehat{p}_{!story}(t - n + 1, .., t + n) \end{aligned} \tag{3.2}$$

When calculated over many (all) potential boundary positions, the result is too noisy to be used directly. We therefore smooth the curve using two moving average (MA) filters; one for short-term (STA) and one for long-term (LTA) averages of the data points. For one half-hour speech segment from TDT-2, the resulting averaged $\widehat{p}_{bound}$ scores were plotted as the two curves in Figure 3.2, where the solid (blue) curve shows the STA values, and the dashed (red) curve the LTA values. We now hypothesize that story boundaries are most likely to occur at positions where the STA has the largest value relative to the LTA and that any story must contain at least one position where the STA is below the LTA. The number of generated boundaries is controlled by the sizes of the MA-filters, which along with the context length n are the parameters for this segmentation method.

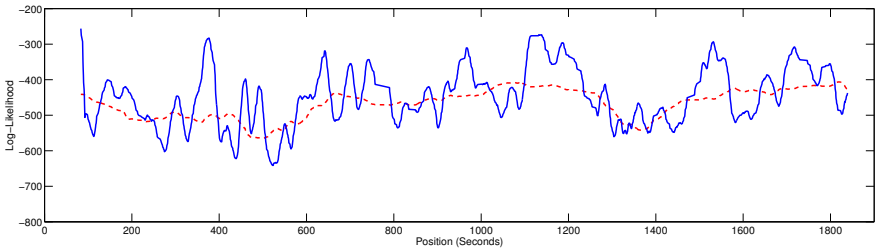

Figure 3.2: *Example of MA-filtered versions of the log-likelihood curve for one 30-minute document from the TDT-2 collection. The solid line represents the short-term, the dashed line the long term average.*

3.2.4 Query-specific Dynamic Segmentation Algorithm

The retrieval task in the TREC8 and TREC9 SDR (SU condition) benchmarks did not necessarily involve creating story boundaries, since the requirement was only to generate positions of relevant passages in the collection. Most participating teams generated explicit story-boundaries using a fixed-duration segmentation, but indexation for IR does not necessarily require a full segmentation. For example, one could index at the document+position level and determine story boundaries ad-hoc, during retrieval, based on queries. Although retrieval in this manner is expected to be much slower than when using a traditional index, the generally small size of many spoken document collections makes this approach perfectly feasible for many practical collections.

Our proposal for a Query-based Dynamic Segmentation Algorithm (QDSA) does not generate a single set of boundaries, but rather creates them on-the-fly, based on the specific information need that is expressed in a query. Retrieval units are identified after the query has been posed, and only those parts of the collection that contain query terms need to be segmented from the rest of the collection.

The implementation we use in our experiments is (almost) a baseline approach to such a scheme: if the distance between 'hits' (positions in the collec-

tion where a query-term is found) is smaller than a certain threshold, they are assumed to originate from a single story. Our basic approach is implemented as follows: first find all positions in a speech transcript at which a query term occurs, then combine all matching positions which are within a certain minimum distance of each other to form the retrievable segments (or stories). All stories generated in this manner must by definition start and end with a query term.

In order to use these segments as stories with a $bm25$ (see Equation 2.8) ranking function, we require for each retrievable segment: the count of term t in story d ($c_{t,d}$), the number of stories containing term t (c_t), the number of terms in story d (c_d), the average story length in the collection (c_D), and the total number of stories in the collection (N). The dynamically generated stories provide $c_{t,d}$, c_t, and c_d, and in our implementation we use the average length of the newly generated stories for c_D, whereas N is obtained by dividing the total length of the collection by c_D.

This baseline approach may not work well when queries are long, or when multiple, adjacent and distinct stories are all relevant, as there may be a very large number of occurrences of query-terms in the collection, leading to excessively large stories being generated. The TDT-2 collection that we used for our experiments intuitively seems relatively suited to this basic approach though. More advanced clustering techniques may be needed for collections and information requests that have multiple distinct relevant stories in close proximity to each other, but as we have no such collections available for testing, we do not investigate this further.

In contrast with the other methods, QDSA is done ad-hoc, during retrieval. As a result, speech transcripts cannot be pre-segmented, changing the demands made on indexation and adding some processing time to the retrieval task. This approach may therefore lead to a potentially less efficient system. For the 400-hour TDT-2 collection and for our experiments this did not pose any practical problems, but in large, real-life settings there may be different limitations and this technique may be unsuitable for such conditions. The only parameter needed for the implementation of this method is the minimum distance between individual segments

3.3 Experimental Setup

The goal of the experiments is to find out whether extrinsic evaluation leads to different conclusions on the best choice of automatic segmentation algorithm for SDR, which segmentation method is the most suitable for SDR and collections with little training material available, and what the impact is of automatic segmentation on retrieval performance in an SDR context as compared to a reference segmentation. Extrinsic evaluation is primarily done using (relative) MAP of retrieval results from experiments using artificial and reference segmentation of the reference transcript of the TDT-2 collection. Intrinsic evaluation is based on segmentation cost. The experimental setup itself is similar to the one used in the TREC8 and TREC9 SDR stories unknown (SU) condition, with

positions in the collection being remapped onto the reference segmentation and scored in the traditional TREC manner.

3.3.1 Experiments

All experiments were done on the English part of the TDT-2 Broadcast News speech collection. This collection is neither very recent, nor particularly challenging from a speech recognition point of view, and the 99 queries for which qrels were defined are also relatively easy when compared to more recent efforts in the context of the TREC benchmarks. However, it has a number of properties which makes it very suitable for comparing various automatic segmentation algorithms in a spoken document retrieval context: i. size, the collection contains 400 hours of speech in more than 21,000 reference stories, providing some challenge when it comes to separating the relevant from the non-relevant content, ii. content, the speech is actual broadcast news speech, making its content both realistic and topically varied, and iii. it has been well researched, because this collection was used for both the Topic Detection and Tracking benchmark and for the TREC8 and TREC9 SDR benchmarks, there is a reference transcript and segmentation as well as a number of automatic transcripts and segmentations available for comparison purposes.

The fact that the TREC8 and TREC9 SDR tasks on this collection do not pose as much of a challenge for modern information retrieval systems as some other IR tasks is not problematic. We are not investigating absolute retrieval performance, but are more concerned with how MAP is impacted as a result of alternative story segmentations. A higher initial MAP on a reference segmentation means a relatively large proportion of the relevant stories was found. As the extrinsic quality of a segmentation is expected to be mostly affected by what happens around relevant segments, this means that performance is potentially more affected by alternative segmentations than when a 'harder' task were used for testing.

Retrieval experiments were done using the $bm25$/Okapi ranking function which was discussed in Section 2.2.3. This function has been used extensively in the various TREC campaigns and is well understood. We used a (simplified) version with two variables: $k1$ and b. The former was set at 1.1 for all experiments, whereas the latter was used as a true variable with values ranging from 0 to 1. A value of 0 for b means the $bm25$ ranking function reduces to a $bm15$ ranking function, and a value of 1 implies $bm11$. For $bm25$/Okapi, story boundaries are explicitly used for two purposes: determining token and document frequency of the query terms, and normalizing the relevance weight for story length. The amount of length-normalization is controlled with the b variable. We assume that the length of the automatic segments is suboptimal if the 'optimal' value of b deviates much from the value found for the reference boundaries. We do not use additional retrieval techniques such as document or query expansion.

The importance of length normalization for $bm25$-based retrieval is illustrated in Figure 3.3 by the (solid) blue lines. The MAPs of baseline experiments

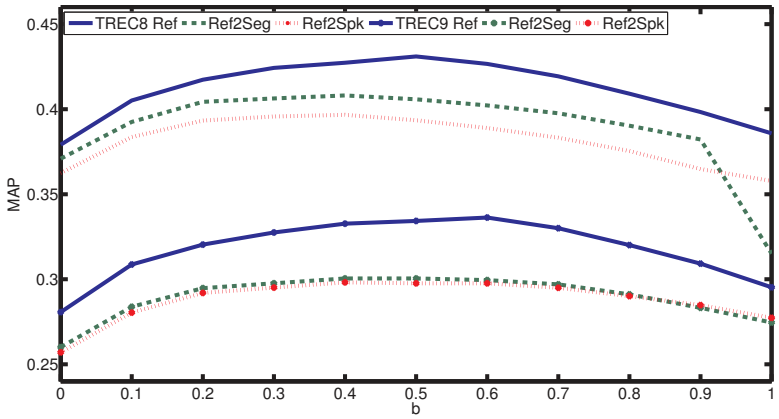

Figure 3.3: *Reference Boundaries - TREC8 & TREC9 queries. MAP when using the reference story segmentation (blue), and the same boundaries mapped to the nearest segment (green, Ref2Seg) or speaker (red, Ref2Spk) boundary. Experiments on the Limsi2008 ASR transcript. The best results were 0.4310@b=0.5 for TREC8 and 0.3363@b=0.6 for TREC9 queries.*

are shown for values of b between 0 and 1, using the reference story segmentation on the 2008 Limsi LVCSR transcript of the TDT-2 collection for the TREC8 (top) and TREC9 (bottom) SDR queries.

3.3.2 Potential Complications

The TextTiling, C99, and WordNet-based automatic segmentation methods need initial basic units for segmentation that are larger than single words. The results of preliminary experiments on using utterance or speaker turns as initial basic units are shown in Figure 3.3. We mapped the reference boundaries to the nearest utterance or speaker turn so as to create an upper limit for performance when using these boundaries as basic units for TextTiling, C99, and WordNet segmentation methods. The dashed (green) lines show that there is a significant reduction in performance when segments are forced to correspond to utterance boundaries, potentially severely limiting performance of these three methods. As can be concluded from the dotted (red) lines, using speaker turns as initial basic units results in worse performance. We therefore use utterances as initial units for our implementations of TextTiling, C99, and WordNet-based segmentation.

Another potential performance reduction is caused by the fact that a significant portion of the TDT-2 transcripts were labeled as non-story content. Reference stories spanned 389 hours and 3.77M words, whereas the collection as a whole totaled 559 hours and 5.13M words. As segmentation does not filter non-important content, and only reference stories can be relevant, an additional

170 hours of non-relevant content in the collection changes the task somewhat for the SU condition experiments, reducing performance. To enable a 'fair' comparison of IR performance, we filter non-story content from the results of our experiments (marked 'filtered' in the tables).

3.3.3 Segmentation Cost

Our primary extrinsic evaluation method was MAP, which is described in Section 2.2.2. In addition, we also calculated the segmentation cost (C_{seg}) for the best performing configurations of each segmentation method. This gives an intrinsic evaluation of the accuracy of the generated boundaries. It is calculated using Equation 3.3, where C_{miss} and C_{fa} are the cost of a miss or false-alarm, p_{miss} and p_{fa} the probability of missing a reference boundary or falsely providing one, and p_{seg} the a-priory segmentation probability. We used $C_{miss} = 10$, $C_{fa} = 1$, and $p_{seg} = 0.1$, following [Fiscus and Doddington, 2002], penalizing 'misses' much more than 'false-alarms'.

$$C_{seg} = C_{miss} \times p_{miss} \times p_{seg} + C_{fa} \times p_{fa} \times (1 - p_{seg}) \tag{3.3}$$

3.4 Results

3.4.1 Statistically Motivated IBM Segmentation

As part of the TDT benchmark, participating labs were required to automatically segment transcripts from the TDT-2 broadcast news speech collection. As the task in the TDT benchmark was to do topic detection and tracking, rather than information retrieval, the evaluation of segmentation was done intrinsically using a cost function. As there was ample training material available, most labs used statistically motivated methods for segmentation. We briefly analyze the IBM segmentation as we see this as a typical example of the expected performance of statistically motivated segmentation.

The results are shown in Table 3.1. This segmentation produces a higher number of stories than the reference (#segs column), which is only partly explained by the larger amount of speech that was was segmented (5.13M vs 3.77M words). The average story length (#terms column) is much lower, as is the standard deviation in story length. Removing non-story segments from IR results, improves MAP, as expected.

3.4.2 Fixed Duration Segmentation

We investigate fixed duration segmentation using non-overlapping and 50% overlapping segments. In the latter case duplicates were (automatically) removed from the ranked result list. For the TREC8 queries, the results for non-overlapping segments are shown in Figure 3.4. The x-axis shows segment size, with the various settings for b on the y-axis. MAP is clearly lower for

		b	MAP	C_{seg}	#segs	#terms (sd)
TREC8	Reference	0.5	0.4310		21754	173.4 (195.7)
	IBM	0.4	0.2958	0.64	43090	101.9 (123.2)
	filtered	0.4	0.3296			
TREC9	Reference	0.6	0.3363		21754	173.4 (195.7)
	IBM	0.3	0.2438	0.64	43090	101.9 (123.2)
	filtered	0.3	0.2611			

Table 3.1: *Retrieval results and statistics for the IBM statistically motivated segmentation.*

small segment sizes, but seems to stabilize at a duration which is similar to the average duration of reference segments. The value of b seems uncritical. The optimal duration of the non-overlapping fixed duration segments is 70 seconds, with an optimal value for b at (a low) 0.1, indicating that segment length showed a reduced correlation with relevance when compared to the reference segmentation. This was not surprising given the fact that the segment duration (in seconds) was fixed so the segment length (in words) was determined only by the average speaking rate. The average segment length for 70 second segments was 178 words, which is comparable to the reference segmentation at 173 words, but the standard deviation was 41, which is considerably less than the 196 words of the reference.

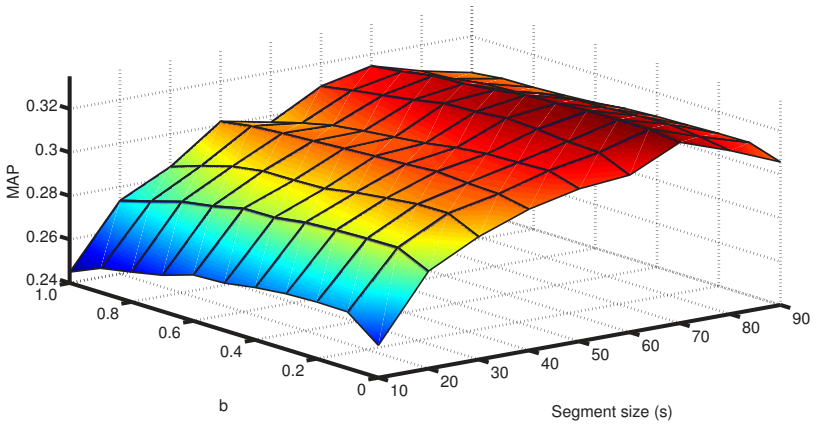

Figure 3.4: *Fixed duration non-overlapping segments - TREC8 queries. MAP of IR using segments with fixed duration. The highest MAP was achieved using 70 seconds and bm25/b = 0.1, resulting in 0.3250 MAP.*

Figure 3.5 shows the retrieval results for the TREC9 queries. As with the TREC8 queries, b is uncritical, but optimal at a rather low value of 0.2. For this set of queries the optimal duration is only 50 seconds. Both 60 and 70

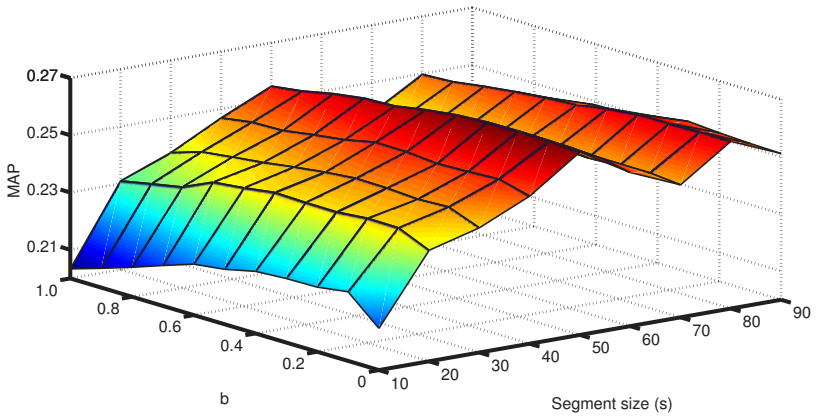

Figure 3.5: *Fixed duration non-overlapping segments - TREC9 queries. MAP of IR using segments with fixed duration. The highest MAP was achieved using 50 seconds and bm25/b = 0.2, resulting in 0.2649 MAP.*

second segments show a rather large drop in performance compared to 50 and 80 seconds, something which is most likely caused by the results on one or more queries being significantly negatively affected by this segmentation, causing some relevant segments to be split, and consequently scored lower than they would otherwise be.

To reduce the likelihood of particularly 'unfortunate' segmentations, one can generate stories with some overlap. Figures 3.6 and 3.7 are the equivalents of Figures 3.4 and 3.5 but with 50% overlapping segments. With this overlap, performance increases to 0.3313 MAP for the TREC8 queries, still at 70 seconds duration, but for the TREC9 queries, the unexpected trough at 60 and 70 seconds from Figure 3.5 has disappeared, and 70 seconds is now also the optimal duration. Although the absolute gain as compared to the non-overlapping 50 second segments is only 0.0032 MAP, the gain compared to the non-overlapping 70 second segments is a more worthwhile 0.0208 MAP.

Table 3.2 contains an overview of the results and statistics on the number of segments, and average segment length and standard deviation for fixed duration segmentation at the duration and b settings that had the highest MAP. Performance is better than the IBM segmentation for MAP, but C_{seg} is much higher (=worse) for fixed-duration segmentation.

3.4.3 TextTiling

The TextTiling automatic story segmentation algorithm takes two parameters: sliding window size and step size. The former controls the amount of context that is considered and the latter influences the size of the segments that are produced. Figures 3.8 and 3.9 show the results for the TREC8 and TREC9

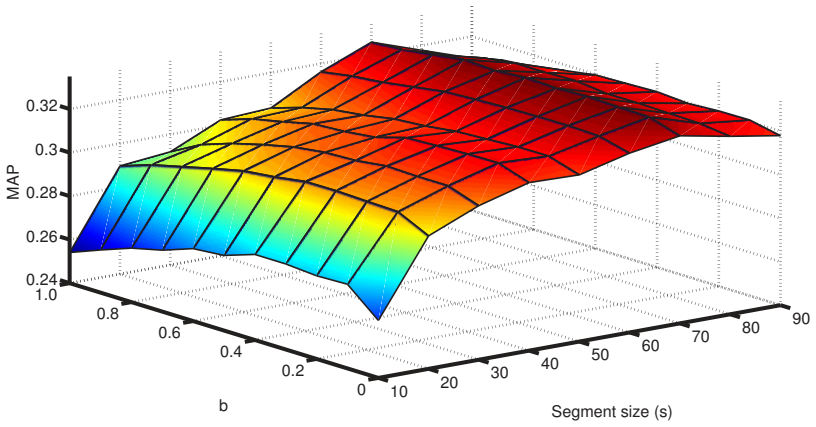

Figure 3.6: *Fixed duration overlapping segments - TREC8 queries. MAP of IR using segments with fixed duration and 50% overlap. The highest MAP was achieved using 70 seconds and bm25/b = 0.3, resulting in 0.3313 MAP.*

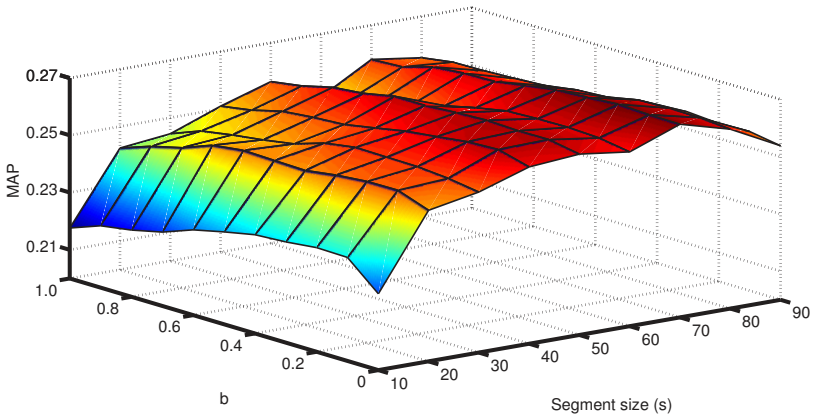

Figure 3.7: *Fixed duration overlapping segments - TREC9 queries. MAP of IR using segments with fixed duration and 50% overlap. The highest MAP was achieved using 70 seconds and bm25/b = 0.2, resulting in 0.2681 MAP.*

queries of IR experiments using the TextTiling segments that were produced with a sliding window size of 10, 20, and 30 words, and a step size of 30, 40, and 50 words (xx and yy respectively in the xx/yy markings on the x-axis). The ideal value for the step size (segment size) parameter is nearly the same for the two query sets; the optimal setting for one set costs around 0.01 MAP on the other queries. For reasonable settings of the TextTiling parameters, b seems uncritical. The best performance for TREC8 queries is found using a setting of

		b	MAP	C_{seg}	#segs	#terms (sd)
TREC8	Reference	0.5	0.4310		21754	173.4 (195.7)
	No overlap 70sec	0.1	0.3250	1.50	28740	178.4 (41.5)
	filtered	0.8	0.3525	1.48		
	50% overlap 70sec	0.3	0.3313	1.29	57109	178.1 (42.4)
	filtered	1.0	0.3616	1.28		
TREC9	Reference	0.6	0.3363		21754	173.4 (195.7)
	No overlap 50sec	0.2	0.2649	1.42	39887	128.5 (30.3)
	filtered	0.3	0.2830	1.40		
	50% overlap 70sec	0.2	0.2681	1.29	57109	178.1 (42.4)
	filtered	0.4	0.2845	1.28		

Table 3.2: *Retrieval results and statistics for fixed duration segmentation at optimal settings.*

30 for both parameters, resulting in a MAP of 0.3388, slightly higher than the best result found using fixed duration segments. For the TREC9 queries, the best performance was achieved at a sliding window size of 40 words and a step size of 30, resulting in a MAP of 0.2639, slightly lower than using fixed-duration segments.

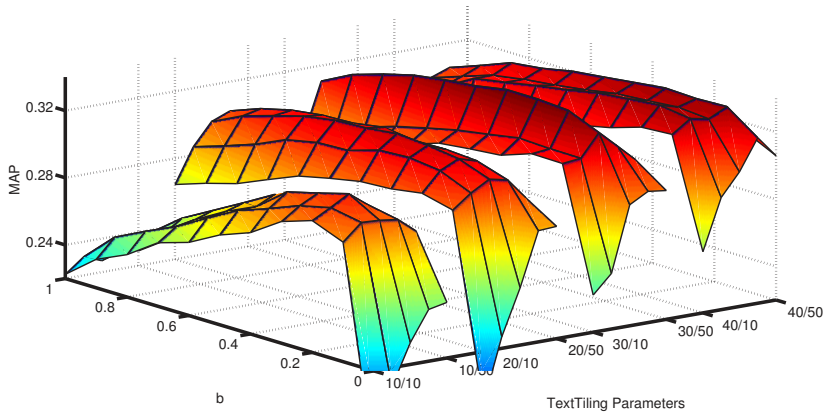

Figure 3.8: *TextTiling - TREC8 queries - Unstopped. MAP of IR using TextTiling segments, parameters: window/step size. The highest MAP was achieved at 30/30 and bm25/b = 0.4, resulting in 0.3388 MAP.*

TextTiling segments documents based on repetition of terms. The common implementation is to filter stop words before doing TextTiling as stop words may show as much repetition between stories as within stories. Figures 3.10 and 3.11 show the retrieval results for the same circumstances as Figures 3.8 and 3.9, but this time applied to the transcripts with stop words removed beforehand. The

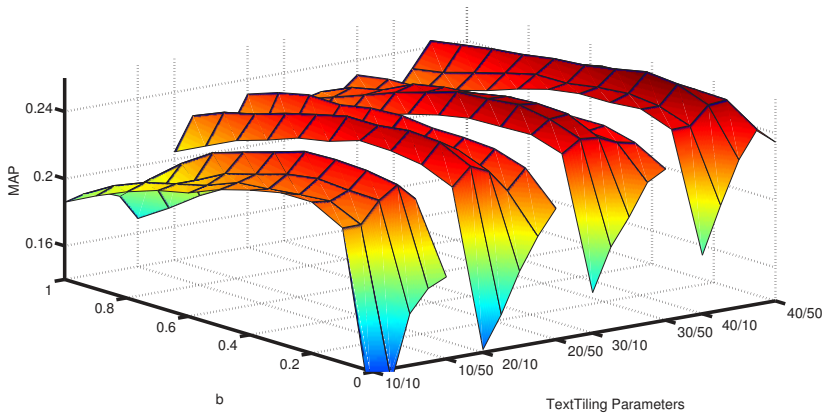

Figure 3.9: *TextTiling - TREC9 queries - Unstopped. MAP of IR using TextTiling segments, parameters: window/step size. The highest MAP was achieved at 40/30 and bm25/b = 0.3, resulting in 0.2639 MAP.*

optimal value of the parameters is slightly lower, which was expected as stop words were removed from the transcripts and the same settings produced a lower number of segments in the case of stopped transcripts. The MAP at the optimal settings however, is almost identical to when TextTiling was applied to the unstopped transcripts. The presence of stop words in the transcript was therefore found not to impact the performance of TextTiling from a retrieval perspective.

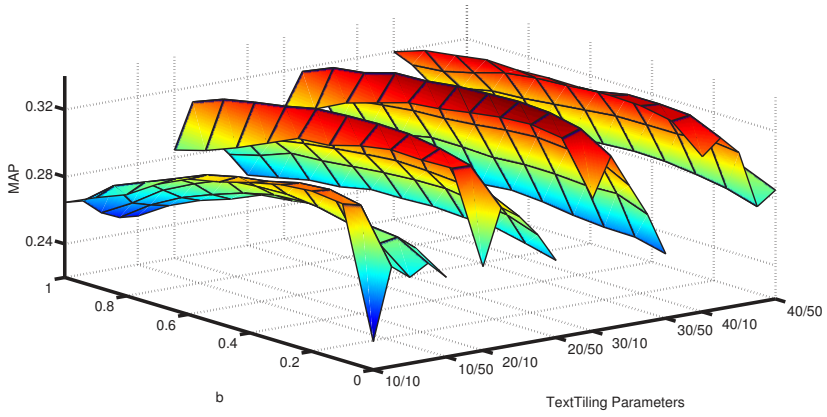

Figure 3.10: *TextTiling - TREC8 queries - Stopwords removed. MAP of IR using TextTiling segments, parameters: window/step size. The highest MAP was achieved at 30/20 and bm25/b = 0.3, resulting in 0.3385 MAP.*

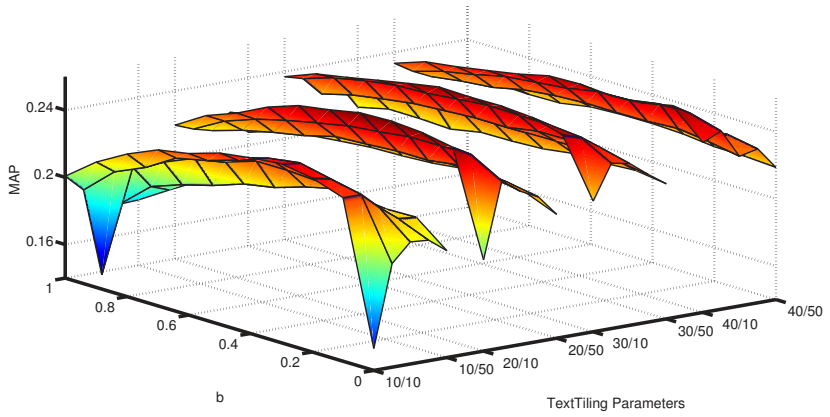

Figure 3.11: *TextTiling - TREC9 queries - Stopwords removed. MAP of IR using TextTiling segments, parameters: window/step size. The highest MAP was achieved at 20/10 and bm25/b = 0.3, resulting in 0.2606 MAP.*

Table 3.3 contains an overview of the results and statistics on the number of segments, and average segment length and standard deviation for TextTiling segmentation at the parameter and bm25/b settings that resulted in the highest MAP. Since the TextTiling algorithm requires an initial segmentation, the maximum performance is somewhat limited by the quality of this initial segmentation. The lines labeled 'Ref2Seg' indicate a segmentation where the reference boundaries are mapped to the nearest utterance boundary and gives an upper limit to what can be expected from segmentation that uses these utterances as basic segmentation units. Segmentation cost of these boundaries is 0.53. In spite of this limitation, the TextTiling algorithm managed to outperform the fixed-duration segmentation *and* the IBM boundaries. Perhaps performance can be further improved through the use of better initial segments, which may require some tuning of the LVCSR system. We did not pursue this in our experiments as this is mainly an ASR optimization issue. Segmentation cost of TextTiling is, depending on the settings used, comparable to that of fixed-duration segmentation.

3.4.4 C99

The C99 segmentation algorithm requires only the mask size as a parameter. The number of generated segments follows from the properties of the text/transcript, so this does not need to be controlled. In practice this method generated less boundaries than the TextTiling approach or the reference segmentation, with around 9k segments for a mask size of 5, and around 12k for a mask size of 17 and up.

Performance of C99 for the TREC8 and TREC9 queries is shown in Figures 3.12 and 3.13 when used on a stemmed transcript. Applying the algorithm on

		b	MAP	C_{seg}	#segs	#terms (sd)
TREC8	Reference	0.5	0.4310		21754	173.4 (195.7)
	Ref2Seg	0.4	0.4081	0.53	20690	194.9 (209.2)
	Unstopped 30/30	0.4	0.3388	1.32	26903	194.2 (164.3)
	filtered	0.3	0.3656	1.32		
	Stopped 30/20	0.3	0.3385	1.31	13876	369.9 (160.1)
	filtered	0.3	0.3641	1.29		
TREC9	Reference	0.6	0.3363		21754	173.4 (195.7)
	Ref2Seg	0.5	0.3005	0.53	20690	194.9 (209.2)
	Unstopped 40/30	0.3	0.2639	1.29	26203	198.2 (139.1)
	filtered	0.3	0.2805	1.28		
	Stopped 20/10	0.3	0.2606	1.16	37217	141.6 (156.2)
	filtered	0.3	0.2761	1.16		

Table 3.3: *Retrieval results and statistics for TextTiling segmentation at optimal settings.*

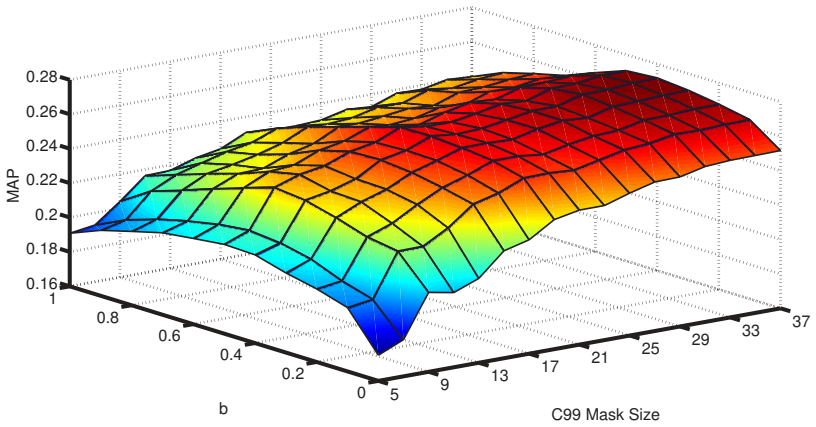

Figure 3.12: *C99 - TREC8 queries - Unstopped. MAP of IR using C99 segments. The highest MAP was achieved at a mask size of 37 and bm25/b =* 0.4*, resulting in 0.2720 MAP.*

a version of the stemmed transcript with stop words removed results in slightly better performance, see Figures 3.14 and 3.15. At their respective optimal settings, C99 performance is worse than TextTiling, worse than using (overlapping) fixed size segments, and worse than the IBM segmentation.

Table 3.4 contains an overview of the results and statistics on the number of segments, and average segment length and standard deviation for C99 segmentation at the mask size and b settings that resulted in the highest MAP. The C99 segmentation algorithm clearly under-segmented the transcripts, with average segment lengths of more than double those of the reference segmentation.

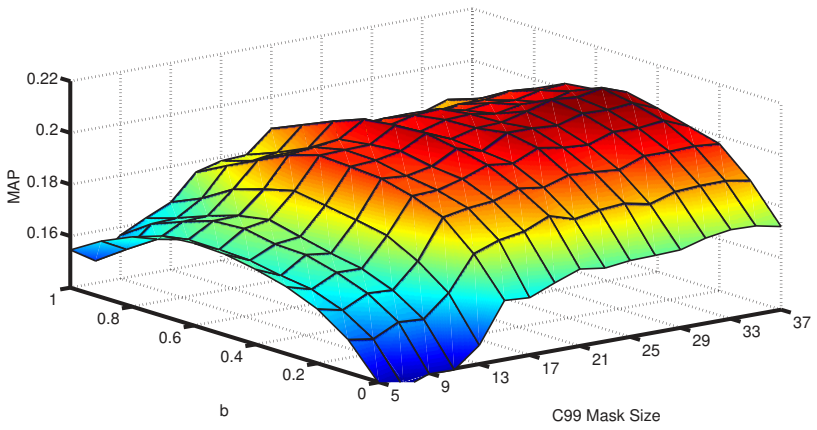

Figure 3.13: *C99 - TREC9 queries - Unstopped. MAP of IR using C99 segments. The highest MAP was achieved at a mask size of 35 and bm25/b = 0.5, resulting in 0.2095 MAP.*

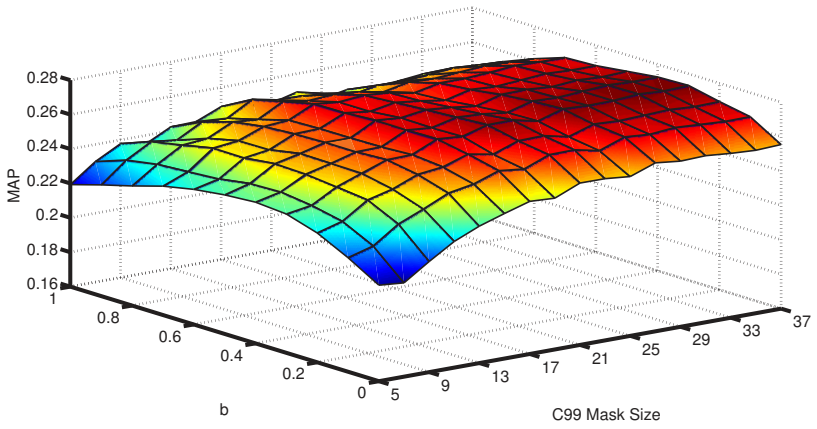

Figure 3.14: *C99 - TREC8 queries - Stopwords removed. MAP of IR using C99 segments. The highest MAP was achieved at a mask size of 33 and bm25/b = 0.3, resulting in 0.2736 MAP.*

3.4.5 WordNet-based Segmentation

Our WordNet-based segmentation algorithm uses the amount of context and the sizes of two moving-average filters as parameters. The latter are used to control the amount of segments that are generated. The results of IR experiments using several values for these parameters are shown in Figures 3.16 and 3.17. On the x-axis, the numbers indicate settings for the three parameters: context size/ma

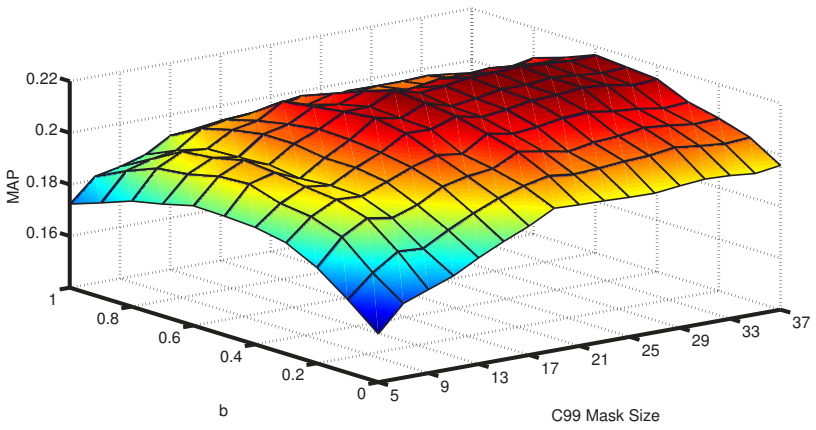

Figure 3.15: *C99 - TREC9 queries - Stopwords removed. MAP of IR using C99 segments. The highest MAP was achieved at a mask size of 31 and bm25/b = 0.6, resulting in 0.2178 MAP.*

		b	MAP	C_{seg}	#segs	#terms (sd)
TREC8	Reference	0.5	0.4310		21754	173.4 (195.7)
	Ref2Seg	0.4	0.4081	0.53	20690	194.9 (209.2)
	Unstopped 37	0.4	0.2720	1.40	12440	412.2 (370.3)
	filtered	0.4	0.2917	1.34		
	Stopped 33	0.3	0.2736	1.35	12182	420.9 (343.5)
	filtered	0.4	0.2926	1.29		
TREC9	Reference	0.6	0.3363		21754	173.4 (195.7)
	Ref2Seg	0.5	0.3005	0.53	20690	194.9 (209.2)
	Unstopped 35	0.5	0.2095	1.40	12416	413.0 (372.3)
	filtered	0.5	0.2250	1.34		
	Stopped 31	0.6	0.2178	1.35	12209	420.0 (340.6)
	filtered	0.6	0.2346	1.29		

Table 3.4: *Retrieval results and statistics for C99 segmentation at optimal settings.*

filter 1/ma filter 2.

No clear pattern emerges as to the optimal setting of the context parameter (the x-axes in the figures), suggesting that the resulting boundaries may be more or less random from an IR point of view. Filter settings that result in a higher number of stories generally also result in higher MAP, but performance remains below TextTiling and also below fixed-duration segmentation. WordNet-based segmentation results in higher MAP than C99 and the IBM segmentation though. The WordNet-based automatic segmentation method we evaluated in these experiments shows little promise for use in spoken document

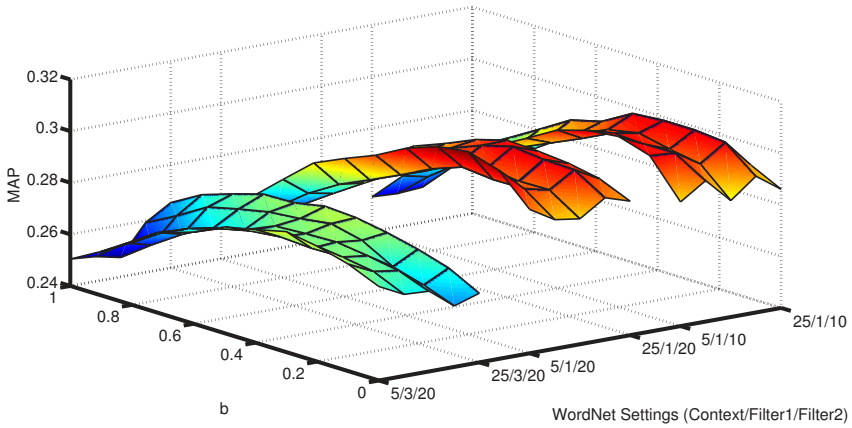

Figure 3.16: *WordNet - TREC8 queries. MAP of IR using WordNet-based segments. The highest MAP was achieved using a context of 5 words, with MA-filters of 1 and 20 samples, and bm25/b = 0.2, resulting in 0.3109 MAP.*

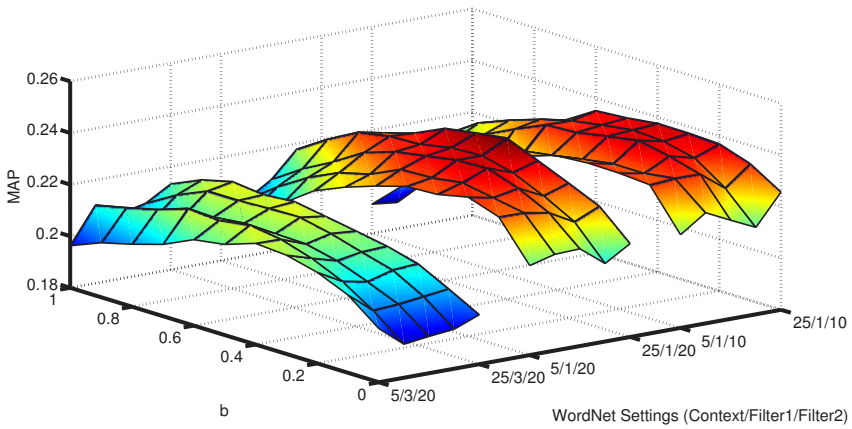

Figure 3.17: *WordNet - TREC9 queries. MAP of IR using WordNet-based segments. The highest MAP was achieved using a context of 25 words, with MA-filters of 1 and 20 samples, and bm25/b = 0.4, resulting in 0.2425 MAP.*

retrieval, as its implementation is relatively complicated due to the need for using WordNet, and the need to optimize three parameters that don't seem to converge in an obvious manner. Table 3.5 has more details on results at the optimized settings for the WordNet-based segmentation algorithm.

		b	MAP	C_{seg}	#segs	#terms (sd)
TREC8	Reference	0.5	0.4310		21754	173.4 (195.7)
	5/1/20	0.2	0.3109	1.47	25376	199.0 (152.0)
	filtered	0.2	0.3330	1.47		
TREC9	Reference	0.6	0.3363		21754	173.4 (195.7)
	25/1/20	0.4	0.2425	1.49	19988	252.6 (171.0)
	filtered	0.4	0.2557	1.49		

Table 3.5: *Retrieval results and statistics for WordNet-based segmentation at optimal settings.*

3.4.6 Dynamic Segmentation (QDSA)

For our proposal for a dynamic segmentation of speech transcripts, the main tunable parameter is the minimum distance between individual stories: all segments that contain query-terms that have less than this specified distance between them are grouped together. Another parameter for this algorithm is the query itself, which is what makes this a dynamic segmentation algorithm. Figures 3.18 and 3.19 show the results of retrieval experiments with various settings for the distance parameter.

The QDSA results in the highest MAP of all the methods we tried in our experiments, even though the resulting (average) segment length is clearly not very accurate as can be seen from the large drop in performance from using values for b of more than 0.1. For a setting of bm25/$b = 0$ (which is effectively $bm15$), QDSA gives similar performance as the reference segmentation at the same setting of b. Comparing like-for-like, by filtering out all segments which are not in the reference segmentation, so for which there is no relevance judgement available, and for a $bm15$ ranking algorithm, the QDSA method resulted in a MAP of 0.3758 for TREC8, and 0.2744 for TREC9, compared with the reference segmentation at 0.3791 and 0.2807 respectively. We therefore conclude that the inclusion of non-story segments and incorrect estimation of segment length are the main reasons for the performance difference between QDSA and manual segmentation in the context of our experiments.

In Table 3.6 the results for the optimal settings of QDSA are shown. The '#segs' column shows the average number of segments generated for each topic, hence the difference between TREC8 and TREC9 queries for the same 30 second setting of the distance parameter. As it happens, the TREC9 queries tend to result in longer segments, indicating that the query terms in those queries more often occur within a 30 second radius from each other. Segmentation cost cannot be easily calculated for this method, as the segmentation is different for each query.

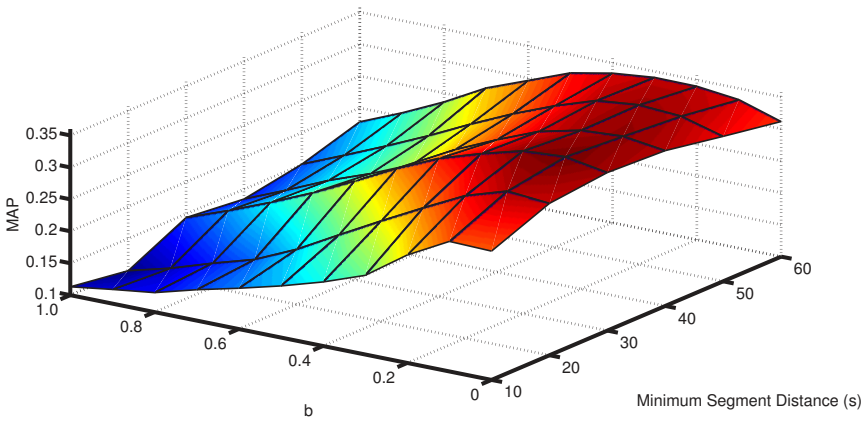

Figure 3.18: *QDSA - TREC8 queries. MAP of IR using QDSA segmentation. The highest MAP was achieved using a distance setting of 30 seconds and bm25/b = 0.1, resulting in 0.3590 MAP.*

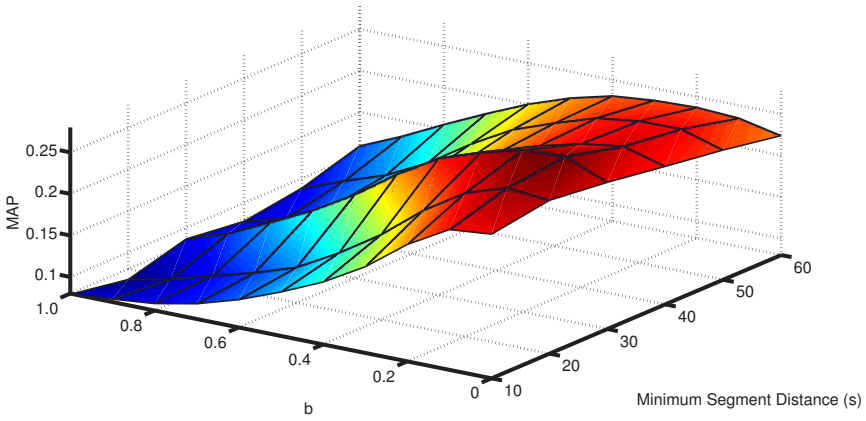

Figure 3.19: *QDSA - TREC9 queries. MAP of IR using QDSA segmentation. The highest MAP was achieved using a distance setting of 30 seconds and bm25/b = 0.1, resulting in 0.2777 MAP.*

3.5 Conclusion

The aim of the experiments in this chapter was to determine whether artificial story boundaries for SDR should be evaluated intrinsically or extrinsically, which method should be used when making an SDR system, and how much retrieval performance is affected by the use of artificial boundaries. We answered these questions by comparing several automatic methods of story segmentation to a

		b	MAP	#segs	#terms (sd)
TREC8	Reference	0.5	0.4310	21754	173.4 (195.7)
	30sec	0.1	0.3590	5473	26.3 (68.2)
	filtered	0.1	0.3902		
TREC9	Reference	0.6	0.3363	21754	173.4 (195.7)
	30sec	0.1	0.2777	6983	45.8 (107.6)
	filtered	0.1	0.2959		

Table 3.6: *Retrieval results and statistics for QDSA segmentation at optimal settings.*

reference segmentation on a fairly large (400h) collection of speech. Only the statistically motivated IBM segmentation required manually labeled training material, all others relied exclusively on properties that were present in the transcript itself. All segmentations were, where possible, evaluated intrinsically using C_{seg}, and extrinsically using a bm25-based IR system and MAP.

Our results show that intrinsic evaluation provides very little information about the usefulness of the segmentation in the context of spoken document retrieval. We therefore conclude that for the segmentation methods that were investigated in this chapter, a cost function is not a suitable target when optimizing segmentation of ASR transcripts for use in SDR.

The worst performing of the methods we investigated were C99 and the WordNet-based method. The former performed much worse than the rest, which is possibly because this method tended to under segment, and we had no method to control the number of segments that were produced. The WordNet-based method also performed worse than the best systems, but was not very far behind. With some further optimization, for example by using a different WordNet similarity measure, and some additional tweaks to parameter settings, or by using better initial segmentation units, it may be possible to increase performance of this method. We did not investigate this any further as we felt the baseline performance that we obtained using this method did not warrant a large amount of additional effort in optimization.

Out of all the segmentation methods we investigated in this chapter, the dynamically generated boundaries of QDSA resulted in the highest MAP. When story length was ignored, thereby effectively retrieving using $bm15$ rather than $bm25$ ranking, the best QDSA configuration was equivalent in MAP to the reference segmentation. If story length is included, QDSA still performs better than the other methods, but is inferior to the reference segmentation. The main drawback of QDSA is that story segmentation must be done on-the-fly, as it is specific for the information request. As such, its implementation is likely to remain much less computationally efficient, and makes it difficult to use traditional indexing methods. This downside may be manageable on many SDR collections and applications, but may be severely limiting for others. Another potential downside is that the method may not work as well on other collections and information requests due to the simplicity of the approach in clustering

query terms into stories. Because of the high performance we obtained, there was little reason to investigate this issue further on the TDT-2 collection.

We were somewhat surprised by the fact that fixed duration segmentation and TextTiling resulted in such similar retrieval performance. Both scored only slightly behind QDSA, but could be implemented in a relatively straightforward and efficient manner. TextTiling my be preferred not only for its IR performance, but also because it may be easier to improve upon by using better initial segments. The fixed-duration segmentation is extremely easy to implement, and when overlapping segments are used, it provides performance that is relatively robust with fewer parameters to optimize. For example, comparing Figures 3.6 and 3.8 shows that performance as a function of segment size and b using fixed duration segmentation is relatively predictable. In conclusion, the fixed duration segmentation with overlapping segments is a good choice for automatic segmentation for SDR, but TextTiling may have slightly more potential when some improvements on initial segments can be made.

3.5.1 Research Questions

Does extrinsic rather than intrinsic evaluation of artificial story boundaries lead to different conclusions with regards to the best choice of segmentation method?
Answer: Yes. Intrinsic evaluation indicated that the best automatic segmentation that we investigated was produced by the statistically motivated IBM system, whereas MAP of retrieval using these segments was the worst except for C99 segmentation. For QDSA, C_{seg} could not be calculated, but this method resulted in the highest MAP of all tested automatic segmentations. For most methods, the optimal parameter settings for MAP did not coincide with minimization of C_{seg} making the intrinsic evaluation also unsuitable for parameter optimization for story segmentation for SDR.

Which is the best method for automatic story segmentation without using additional resources in the context of SDR, based on extrinsic evaluation?
Answer: The highest MAP was achieved using QDSA. However, our collection may have been especially suitable for this method due to the shortness of the reference segments and the expected high contrast between adjacent segments. TextTiling and fixed duration segmentation resulted in similar IR performance, but TextTiling performance may have been limited by the use of utterances as initial segments. The fact that the utterances we used were based only on silences in the speech, means that it may be possible to improve TextTiling performance by using more advanced utterance boundary detection techniques at the ASR level. This makes TextTiling our preferred method when QDSA is unpractical due to collection size or other properties of the collection not favoring the simple clustering approach we used.

What is the impact of artificial story boundaries on retrieval performance of an SDR system when compared to a reference segmentation?
Answer: The effect of using artificial story boundaries, as generated using the

methods that we investigated in this chapter, on MAP is relatively severe. The best system (QDSA) resulted in a relative reduction in MAP of $\sim 10\%$, and the second-best system (TextTiling) in a reduction of $\sim 15\%$. The worst system we tested resulted in a reduction in MAP of $\sim 33\%$.

3.5.2 Summary

Information retrieval requires that a collection is in a computer readable (typically textual) form, and that it is divided into 'natural' retrieval units, typically coherent stories. For most textual collections these requirements are easily met, as individual stories are often marked as chapters or paragraphs. In the case of spoken document retrieval, raw speech samples are typically converted into text by an ASR system, but the resulting transcripts rarely contain clues on story boundaries. We therefore need to segment ASR transcripts into coherent stories in order to be able to index them for IR.

In this chapter we have investigated generating artificial story boundaries (segments) in an SDR context. We have shown that intrinsic evaluation of automatic segmentation does not correlate well with extrinsic evaluation: the method that resulted in the lowest (=best) C_{seg}, also resulted in a low (=bad) MAP when used in retrieval experiments. Ideally a segmentation would score well on both, as this means the boundaries are similar to the references and also good for retrieval purposes. The former is mainly important for presentational purposes, as a system that presents only complete and coherent fragments as results is expected to be preferable for a user to one that provides a 'random' position somewhere within each fragment. On the other hand, user preferences may depend on many other factors that we could not take into account in our investigations, so we did not include this as a parameter in our experiments. As we focused on 'core' performance, we felt that extrinsic evaluation was the better choice for automatic story segmentation for SDR.

Out of the automatic segmentation methods that we tested, the best (extrinsic) performance was achieved using the QDSA (dynamic) segmentation algorithm. The main downside of this approach is that it makes indexing (much) less efficient, at least in our implementation. For SDR on large collections with many users this may become a real bottleneck and make the use of this method unfeasible. It is also unclear how well this approach would work on collections that are less well suited to the particulars of the algorithm. As a general recommendation, it is therefore probably better to use either TextTiling or fixed duration segmentation, where the former may be especially beneficial for performance when there is an initial segmentation, for example in utterances, that coincides well with true story boundaries.

4

Speech Transcript Evaluation

In this chapter various methods for the evaluation of automatic speech transcripts for use in the context of spoken document retrieval are investigated. Whereas traditional speech transcript evaluation is typically performed in a dictation-type context with a focus on counting errors that need correction, in the context of SDR the focus should be on the consequences of transcript errors on the search results.

Spoken document retrieval is usually implemented as a customized information retrieval engine on the output of a Large Vocabulary Continuous Speech Recognition (LVCSR) system. Due to various practical and theoretical limitations, ASR inevitably comes with transcript noise that may compromise retrieval performance. The typical challenge for dictation-type ASR applications is minimizing the number of errors given a certain speed target, hence evaluation using word error rate, see Equation 2.1. When using such transcripts in an SDR context, ASR errors may result in a retrieval bias, especially when errors occur for query terms. As it is not possible when using only WER to differentiate between the words on which the errors occur. As WER is based on a simple count of errors, it may not be adequate for detecting performance issues that are dependent on the role of words in a retrieval context.

The work in this chapter is motivated by the need for measuring retrieval bias that is caused by transcript noise. Whereas WER is an intrinsic evaluation metric for ASR transcripts, a TREC-style approach to IR evaluation using MAP provides an extrinsic measure of transcript quality. However, MAP is not very practical for ASR transcript evaluation due to its dependence on qrels (see Section 2.2.2), and has also been shown to be relatively robust towards transcript errors [Garofolo et al., 2000b]. We therefore need an alternative approach to extrinsic evaluation of speech transcripts that does not require qrels, but provides the same information as can be learned from using (relative) MAP.

The quality of an ASR transcript is often strongly correlated with properties of the speech, such as its type (spontaneous, rehearsed), its acoustic properties (noisy or clean conditions), and its similarity to the acoustic models, and somewhat with its content through linguistic models. An optimal configuration for broadcast news transcription is unlikely to perform as well on spontaneous conversational speech. Various parameters, such as the choice of acoustic and

linguistic models, their relative weights, penalties related to word length, and word insertion penalties can all be adjusted to fit the task. The optimal values for these parameters are set based on the outcome of an evaluation. For dictation applications, WER may be a suitable target, but a transcript for use in a search application may benefit from a different optimization, for example one that favors information carrying content over filler words.

Evaluating a literal orthographic transcript when it is used in SDR is potentially suboptimal because the IR process itself does not use this literal transcript, but operates on the result of an indexation process: a process which typically includes removing stop-words (highly frequent terms with low informational content), and applying a stemmer such as a Porter stemmer [Porter, 1980]. Furthermore, indexation usually treats a collection as bags-of-words and the IR ranking algorithm may process word frequency in a non-linear manner [Spärck-Jones et al., 2000]. Ranking stories for expected relevance implies a comparison between stories in a collection, making not just the properties of a story, but also its properties in comparison to other stories the basis for retrieval. Evaluation of an ASR transcript based on a simple word-for-word comparison therefore potentially misses many aspects which are of great importance to retrieval performance, while simultaneously overemphasizing aspects which can be safely ignored.

In this chapter we investigate several alternatives to WER and MAP for the evaluation of automatic transcripts in an SDR context. These methods include weighted error rates, and rank correlation and overlap-based measures on IR results. A distinction is made between intrinsic and extrinsic approaches to ASR transcript evaluation. Our aim is to establish methods for extrinsic evaluation that show a high correlation with relative MAP, but without requiring qrels. In addition, we investigate intrinsic approaches that are a potential alternative for WER and may be used if for any reason extrinsic approaches provide unsatisfactory results, or cannot be applied for practical reasons. The research questions that we intend to answer are:

- Can we evaluate ASR transcripts in an intrinsic manner that is more appropriate for SDR than traditional WER?
- Which method for extrinsic evaluation has the highest correlation with relative MAP?
- Can extrinsic evaluation of ASR transcripts without qrels be reliably used to predict relative MAP?

This chapter is organized as follows: Section 4.1 provides an overview of earlier work on transcript evaluation in the context of spoken document retrieval, after which Section 4.2 introduces the methods we tested in our experiments. Section 4.3 describes the experimental setup for our investigation into the correlation between our ASR-for-SDR methods and MAP, the results of which are given in Section 4.4. Finally, Section 4.5 provides a conclusion and explicit answers to our research questions. Most of the work in this chapter was published in [van der Werff and Heeren, 2007] and [van der Werff et al., 2011].

4.1 Previous Work

One of the earliest formal large-scale investigations into SDR was done for TREC7 SDR [Garofolo et al., 1998], which was a spoken document retrieval benchmark held in the context of the Text REtrieval Conference (TREC) of 1998. Several tasks were defined, including retrieval on a reference transcript, retrieval on self-produced ASR transcripts, and retrieval on ASR transcripts from other participants. Story boundaries were provided by the organization and evaluation of SDR was done in the same manner as most other TREC tasks, using primarily MAP [Voorhees and Harman, 1998].

For TREC7 SDR, a high correlation (0.87) was found between the ranking of systems based on WER and MAP, indicating that intrinsic and extrinsic evaluation resulted in a similar ranking of systems. Recognizing that this may have been coincidental given the procedures used, several alternative ASR for SDR evaluation methods were investigated. i. (Story) Term Error Rate (TER) is similar to WER but is calculated as the sum of the difference in term counts for each story divided by the total term count of the collection. It does not require an explicit word-level alignment, and a substitution error is counted as an insertion plus a deletion. ii. Stemmed and Stop Word Filtered WER is similar to WER but calculated after the removal of stop words and stemming of all remaining terms, and iii. Named Entity WER is similar to WER but calculated after the removal of all non named entities from the transcripts. Only the latter seemed to result in a slightly better correlation with MAP than WER (0.91), whereas the other measures gave results similar to WER ($\sim$0.85).

In subsequent editions of the TREC SDR benchmarks [Garofolo et al., 2000b] the size of the collection and the number of topics was greatly increased. Cross-system results indicated that performance was only minimally impacted by errors in the automatic transcripts. The average reduction of MAP for each additional percent WER was 0.0016, for systems with WER ranging from 13 to 30%. As MAP of an IR task performed on a reference transcript was very close to one on the best of the ASR outputs, the task of SDR on English language broadcast news speech was declared solved [Garofolo et al., 2000b].

In [Singhal and Pereira, 1999] the relative impact of transcript noise on SDR performance was investigated, where multiple types of errors were recognized. Three types of error were defined: i. the term count was different but not zero in both reference and ASR transcript, ii. the term count was zero in the reference but not in the ASR transcript, and iii. the term count was zero in the ASR transcript but not in the reference. The least impact could be attributed to errors of type i, where only the count of terms was changed but not their binary presence. The complete deletion of a term from a document (type iii) mostly impacted long queries, whereas the insertion of a term that was not present in the speech (type ii) was most detrimental for retrieval using short queries. These results showed that the impact of errors not only depends on the content of the collection but also on the properties of the queries.

The Indicator Error Rate (IER) [Macherey et al., 2003] combined the find-

ings on error types [Singhal and Pereira, 1999] with TER [Garofolo et al., 1998]. It is a variation on TER where both ASR and reference transcripts are first stopped and stemmed, but instead of 'real' counts, binary presence is used. Their experiments showed that correlation between IER and MAP was potentially higher than for WER or TER, but results were inconclusive due to the limited number of data points.

WER, TER, and IER are discussed in more detail in Section 4.2.1.

4.2 Evaluating ASR

The entire process of spoken document retrieval can be broken down into a number of large and small tasks. Figure 4.1 shows the typical steps in an SDR system. Not all SDR systems operate in exactly this manner, but typically a 1-best transcript is story segmented, stopped/stemmed, and then enters the indexation process. Figure 4.1 is intended to clarify the different stages in the SDR process at which ASR evaluation can be performed.

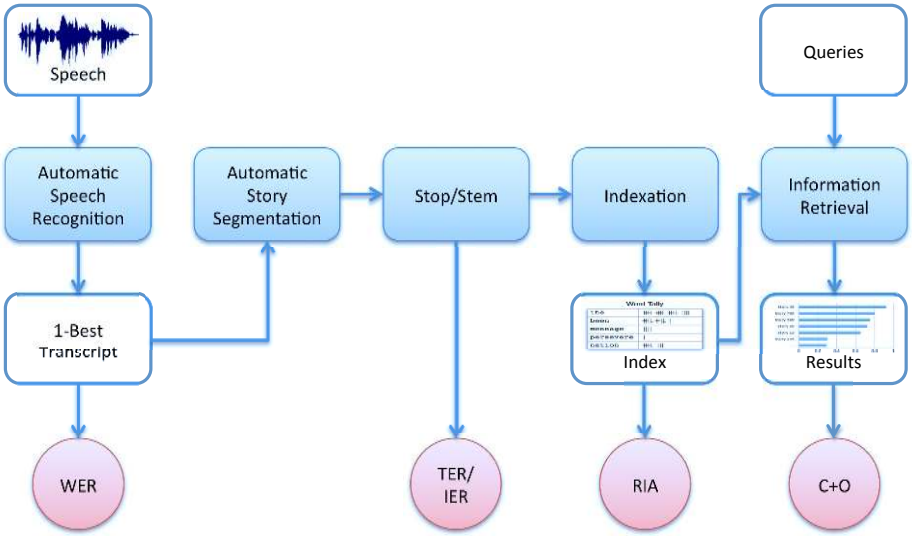

Figure 4.1: *Overview of the steps an ASR transcript takes through an SDR system and potential evaluation points.*

Traditional approaches such as WER and TER operate directly on the 1-best transcript or are calculated after stopping/stemming, see Section 4.2.1. The RIA approach [van der Werff and Heeren, 2007] that we proposed, see Section 4.2.2 includes the indexation process into the evaluation, and the Correlation and Overlap (C+O)-based methods that are discussed in Sections 4.2.3 and 4.2.4 use the results of the IR process for evaluation. This Section provides an overview of the various methods for ASR transcript evaluation that we test for correlation with MAP in our experiments.

4.2.1 Word, Term, and Indicator Error Rate

As explained earlier, the evaluation of 1-best literal transcripts is most commonly done using Word Error Rate (WER), see Equation 4.1. The number of substitutions, insertions and deletions (S, I and D) are determined through a dynamic programming, minimum Levenshtein distance function alignment of reference and hypothesis transcript [Levenshtein, 1966]. This is normalized for the number of words in the reference (N). A literal, human generated, reference transcript is used as ground truth. A word is erroneous if its superficial form differs from the reference, regardless of meaning or nature of the error. In certain cases, such as numbers, capitalization, or compound words, normalization may need to be applied to get the desired assessment of performance.

$$WER = \frac{S + I + D}{N} \times 100\% \tag{4.1}$$

The most frequent type of error in WER-optimized ASR systems is the substitution error. Substitutions are effectively the same as a deletion (of the correct word) and an insertion (of the wrong word), which for WER calculation are counted as single errors, as they are the logical consequence of a one 'mistake' in the transcription process. Indexation for IR typically ignores word order and only registers term counts for each segment or document. With positional information removed, insertions and deletions that were counted as substitution errors are no different from any other error, so there is little reason to count substitution as a single error. When an automatic transcript is used for SDR, it is typically subject to post-processing, such as stop word removal, stemming, and story segmentation. If we assume that any error that has no impact on an index, has no impact on the performance of an SDR system, it makes sense to evaluate ASR quality after all pre-processing has been done.

Term Error Rate (TER) [Garofolo et al., 1998] is a suitable alternative to WER for evaluation of bags-of-words, see Equation 4.2. The count of term w in document d is represented by $A_{w,d}$ and $B_{w,d}$ for the reference and ASR-based transcript respectively. TER can often be approximated from WER by doubling the value for substitutions, making TER highly correlated with WER. Explicitly optimizing for TER may reduce correlation with WER, but may also result in improved performance in the context of SDR. For example, not producing any words at positions with low ASR confidence scores, may increase precision of search results and reduce TER, but is likely to lead to an increased WER.

$$TER = \frac{\sum_d \sum_w |A_{w,d} - B_{w,d}|}{N} \times 100\% \tag{4.2}$$

A method that is closely related to TER, is the Indicator Error Rate [Macherey et al., 2003] which is calculated in a similar manner to TER, see Equation 4.3. Instead of term counts, only binary presence (BPA and BPB) is used and instead of the total term count in the documents, the number of unique terms is used (N_u). This approach is potentially interesting for SDR applications as

a change in binary presence affects retrieval performance more than a simple change in term count [Singhal and Pereira, 1999].

$$IER = \frac{\sum_d \sum_w |BPA_{w,d} - BPB_{w,d}|}{N_u} \times 100\% \tag{4.3}$$

4.2.2 Relevance-based Index Accuracy

Evaluation of ASR transcripts is mostly done in order to enable the optimization of the many variables and parameters of an LVCSR system. The goal is to produce the best, or rather most useful, results. But what is best or most useful is not always closest to the reference in a linear manner. Most IR systems calculate the contribution to the relevance of a story for a query from the frequency of (query) terms in the story. The manner in which this so-called 'relevance weight' is calculated is one of the defining aspects of an IR system. Typically, the relevance weight is related to the term count in a non-linear manner. This non-linearity is ignored when using TER and potentially overemphasized when using IER, but it can be captured accurately using the Relevance-based Index Accuracy (RIA) [van der Werff and Heeren, 2007].

One of the most popular approaches to relevance weight calculation is Okapi /$bm25$ [Spärck-Jones et al., 2000], which we discussed in Section 2.2.3 and can be calculated using Equation 2.8. All terms receive a document and term-specific weight which depends both on the number of occurrences within a document (term frequency) and on the number of documents in the entire collection in which they occur (document frequency). The non-linear interpretation of word counts is implemented using tuning variable k_1. If we assume queries with no duplicate terms, the first part of Equation 2.8 can be omitted, resulting in Equation 4.4.

$$bm25 = \sum_{q=1}^{Q} \frac{c_{t_q,d} \times (k_1 + 1)}{c_{t_q,d} + k_1 \times (1 - b + b \times \frac{c_d}{c_D})} \times log \frac{N - c_{t_q} + 0.5}{c_{t_q} + 0.5} \tag{4.4}$$

In contrast to TER and IER, RIA calculates the similarity between reference and ASR transcript not from the counts of terms and documents, but from the weights which determine the contribution to the expected relevance of a story segment for each term. As $bm25$-based retrieval operates with a simple sum of contributions of each query term, this can be implemented in a fairly straightforward manner. When using the simplified version of $bm25$ from Equation 4.4, contributions of each term to the overall relevance of a document for a query containing this term, are independent of the other terms in the query, and can be pre-calculated for a closed collection. For a given value of b and k_1, we can efficiently calculate $bm25$-weights from only the term counts and collection-wide statistics. As relevance weights are independent of queries, they result in intrinsic evaluation of ASR transcripts.

Comparing the weights between an index based on a reference transcript and one based on an ASR transcript can be done with the Vector Space Model

(VSM) [Salton et al., 1975]. We define N as the total number of unique terms in the collection, including both the automatic transcript and a ground-truth manual reference. A (sparse) vector representation of the collection can then be made which has a size which is equal to N times the number of documents (or stories) in the collection, where each entry $w_{n,d}$ is equal to the $bm25$-weight of term n in document d. Using the VSM, we calculate the angle between two vectors and use its cosine as a measure for similarity between the two, which we call Relevance-based Index Accuracy. It is calculated using Equation 4.5 (a standard VSM calculation), where $\overrightarrow{R}$ and $\overrightarrow{A}$ are the $bm25$-weight-based vectors made from reference and automatic transcript respectively.

$$RIA = \frac{\overrightarrow{R} \cdot \overrightarrow{A}}{\|\overrightarrow{R}\| \times \|\overrightarrow{A}\|} = \frac{\sum_{d=1}^{D}\sum_{n=1}^{N} R_{w_{n,d}} \times A_{w_{n,d}}}{\sqrt{\sum_{d=1}^{D}\sum_{n=1}^{N} R_{w_{n,d}}^2} \times \sqrt{\sum_{d=1}^{D}\sum_{n=1}^{N} A_{w_{n,d}}^2}} \tag{4.5}$$

The main difference between TER and RIA is that the latter doesn't treat all errors equally and more severely penalizes an error that is made on more distinctive terms, something that should not be confused with term importance. Typically terms that are very frequent have low relevance-weights, and errors on such terms result in small changes to those weights, and therefore have less of an impact on RIA.

4.2.3 Rank Correlation of Retrieval Results

Our main criticism of WER in an SDR context is that it may not properly acknowledge the importance of errors. Although TER, IER, and RIA aim to improve this by calculating error rates based on word counts or similarity in term weights, they still amount to intrinsic evaluation where the actual use of the system is not part of the evaluation process. Effectively, when using these methods, the quality of the transcript is assumed to be independent of the needs of users.

In practice, term weights cannot be assumed to be representative of the importance of a term, i.e., a spelling error is likely to be very distinctive for a document, resulting in a high weight, but from a retrieval point of view, such anomalies are rarely of any importance. In fact, term weights are meaningless unless the term plays a role in the retrieval process, and then only in a relative sense when compared to the weights of other terms or of the same term in other documents. It is therefore impossible to determine the quality of a transcript for an IR task, without specifying this task.

Extrinsic evaluation is standard practice for information retrieval, for example using MAP (see Section 2.2.2). With extrinsic evaluation the exact use is part of the evaluation, and in IR this is accomplished by evaluating the quality of a ranked result list of a set of representative information needs. We propose to use a similar approach for evaluation of ASR in the context of SDR. However, instead of using relevance judgements (qrels) as the ground truth, we use the results from a retrieval run on a reference transcript directly. Our approach is best illustrated by Figure 1.3 from Section 1.3.2.

Using rank correlation measures, we can determine the similarity in ranking between two lists containing (the same) documents, typically resulting in a value of 1 for identical lists, 0 for uncorrelated lists, and -1 for lists which are inversely ordered. If two IR tasks are performed in the same manner on almost identical systems, which differ only in their use of a reference or automatic transcript as basis for the indexation, then the differences between the ranked results must be the caused by transcript errors from the ASR process. We therefore expect a comparison between these results to adequately characterize the quality of an ASR transcript in an SDR context.

It is important to realize that one does not need to know the absolute performance of systems to be able to characterize the difference between them. In our approach, we take the retrieval results on a reference transcript as the gold standard, even though it is very unlikely that these are 'perfect' (MAP=1) from a retrieval perspective. In the case of ASR transcript evaluation in an SDR context, one can usually safely assume that performance on a reference transcript is better than on a noisy ASR transcript. On the other hand, it is possible that segments of speech with poor acoustic properties are also less often relevant than clean studio-produced speech. In such cases, the noisy segments are likely to be ranked lower due to ASR errors, which may result in better IR performance. A noisy automatic transcript may therefore sometimes result in better performance than a human-made reference. In general however, we do not consider noisy transcripts to be desirable in an SDR context.

The most commonly used methods of rank correlation are Kendall's τ and Spearman's ρ [Kendall, 1938], which are designed for comparing lists that are identical except for the ranks of their contents. The ranked results from IR are typically limited to the top-n segments, meaning that the results for IR on different representations of the same collection are likely to only overlap partially. This poses some challenges for the implementation, but with a small modification of the calculation one can get a reasonable estimate anyway.

We alter our results lists in the same spirit as was proposed in [Fagin et al., 2003]. We make the assumption that any document which is contained in one of the two lists, but not the other is actually (invisibly) present at position $N+1$, with N the number of elements in the original list. By doing this we obtain two lists which contain the same elements but with a different ordering. We also obtain two lists with potential ties at position $N+1$, which needs to be dealt with separately for each of the rank correlation measures we use.

Kendall's τ Using Equation 4.6 we can calculate Kendall's τ, with n_c the number of concordant pairs (pairs which are in the same order in both lists), and n the number of elements in each list. As tied ranks are neither concordant nor discordant, and following [Fagin et al., 2003], we adapted the counting of concordance by increasing n_c by 0.5 for tied ranks.

$$\tau = \frac{4 \times n_c}{n(n-1)} - 1 \tag{4.6}$$

Spearman's ρ For Spearman's ρ, see Equation 4.7, absolute rank numbers are used rather than concordance. Ties are therefore not problematic, but absolute ranks are important. Our extended lists contain potentially many entries at position $n+1$, which is a rather optimistic estimate of their true ranks. We resolve this by assigning the average position to all elements tied for one rank. For example, when one list has 5 elements not present in the other, these would initially all be assigned rank $n+1$. Averaging their position gives each of them a rank of $n+3$.

$$\rho = \frac{\sum_i (x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_i (x_i - \overline{x})^2 \sum_i (y_i - \overline{y})^2}} \tag{4.7}$$

A potential concern with the use of τ and ρ for rank correlation of ranked result lists from IR runs, is that all positions are treated as equally important. If two documents were swapped for position at the top end of the list, this is not counted any different than a swap at the bottom end of the list. Since IR experiments are typically evaluated for 1000 results, but users rarely ever look this far down the list, there is the potential for too much emphasis being put on the order of documents which are never actually inspected and whose positions have no bearing on subjective quality of the system. We therefore also investigated two alternative rank correlation methods which place more importance on the top end of the result list.

Average Precision inspired τ With Equation 4.8 we can calculate τ_{AP} [Yilmaz et al., 2008], where C_i is the number of items above rank i in one list that are correctly ranked with respect to the item at rank i in the other list. It is easy to see how, similar to Average Precision, a correlation is calculated at each position and this value is finally averaged. It is similar to Kendall's τ in that it uses concordance for calculating rank correlation.

$$\tau_{ap} = \frac{1}{n-1} \sum_{i=2}^{n} \left(\frac{C_i}{i-1}\right) - 1 \tag{4.8}$$

Blest's ω One can calculate Blest's ω [Blest, 2000] using Equation 4.9, where q_i is the rank in the other list of the item at rank i. Blest's ω is somewhat similar to Spearman's ρ in that it uses the distance between the ranks of the same document for calculating rank correlation, but just like τ_{AP} rank differences at the top end of the list are emphasized over those at the tail end.

$$\rho_B = \frac{2n+1}{n-1} - \frac{12}{n(n+1)^2(n-1)} \sum_{i=1}^{n} (n+1-i)^2 q_i \tag{4.9}$$

4.2.4 Overlap of Retrieval Results

Average Overlap As an alternative to rank correlation, Average Overlap (AO) [Wu and Crestani, 2003] can be used to compare lists purely for their

content. The overlap between two sets of documents is a measure for similarity, but overlap, as precision in IR, is a set-based measure. Just as with AP however, it can be expanded to work for ranking, see Equation 4.10, where k is the depth of the evaluation and $|A_{:d} \cap B_{:d}|$ the number of elements that are present in both lists up to depth d.

$$AverageOverlap = \frac{1}{k} \sum_{d=1}^{k} \frac{|A_{:d} \cap B_{:d}|}{d} \tag{4.10}$$

One of the potential advantages of AO over correlation-based measures is that there is no underlying assumption of both results lists containing the same documents. For the correlation-based methods, the resulting value falls in a range of -1 to 1, but the numbers are not always easy to interpret. For example, if the same n documents appear in both top-n lists, one could say that this implies a reasonable similarity, especially for $n < 1000$ and a collection containing 20k+ documents. However, if these are in uncorrelated order or inverse order, the correlation coefficient will be 0 or less, but it is unclear how this then relates to the similarity of the result lists. A rank correlation of 0 may in this case indicate a higher initial similarity than a positive correlation. Average Overlap gives a value between 0 and 1, where 0 means the two lists are disjoint, and 1 means they are identical both in content and ordering. Because AO operates in a similar manner to MAP, it puts more emphasis on the top end of the results lists in the same way as is the case for MAP, increasing the chances of finding a high correlation between AO and MAP.

Rank-Biased Overlap An alternative to AO was developed in Rank-Biased Overlap (RBO) [Webber et al., 2010]. This introduces a variable p that can be used to control the amount of bias that is given to the top end of the result list. RBO is calculated using Equation 4.11, where $p = 0$ means only the topmost result is considered and $p = 1$ means that the evaluation becomes arbitrarily deep.

$$RBO = (1 - p) \sum_{d=1}^{\infty} p^{d-1} \frac{|A_{:d} \cap B_{:d}|}{d} \tag{4.11}$$

With AO it is possible for the results in the tail to dominate those at the top, making it unclear whether the resulting number is truly representative of the differences in ranks or a side-effect of the chosen evaluation depth. This is not the case with RBO as the combination of evaluation depth and p determines the maximum achievable RBO value, which is 1 for an infinitely deep evaluation. This maximum can be found using Equation 4.12. For example, with $p = 0.95$ and an evaluation depth of 1000, the RBO_{max}=1.000, but with a depth of 100, RBO_{max} is reduced to 0.9991 (the remaining 0.0009 is an uncertainty, reserved for results at ranks 101+). For most practical values of p, the typical IR evaluation on 1000 results should be sufficient to get an accurate RBO value.

$$RBO_{max} = 1 - p^k - \frac{k(1-p)}{p} \left(\sum_{d=1}^{k} \frac{p^d}{d} + ln(1-p) \right) \tag{4.12}$$

4.3 Experimental Setup

The goal of the experiments is to find answers to the research questions that were stated in the introduction of this chapter. These are to i. determine how we can intrinsically evaluate ASR transcripts in a manner that better reflects the demands in an SDR context than WER, ii. establish which of our proposed methods provides the best correlation with MAP for transcript noise and story segmentation errors, and iii. find out whether extrinsic evaluations that do not use qrels can be reliably used to predict relative MAP. We do this by testing the various methods for ASR evaluation that were explained in Section 4.2 for their correlation with the relative retrieval performance of an IR system using the automatic transcripts. We look at two aspects of correlation: i. how similar is the ranking of the various transcripts based on these measures to the ranking according to MAP, and ii. can we use the values we obtain from these measures to predict the degradation in MAP score as a result of transcript noise. The former can be established using a rank correlation, the latter using the linear correlation between true relative MAP and its estimation from each of the measures.

We use MAP as our target measure because it was used extensively for the TREC8 and TREC9 SDR benchmarks, and is a de facto standard for IR evaluation in TREC benchmarking. As there are 99 queries and corresponding qrels available for the TDT-2 English language broadcast news speech collection, calculation of MAP was relatively straightforward (unlike for typical ad hoc retrieval tasks). Our choice for MAP does not mean that we feel it is necessarily the best or only way of measuring the impact on IR of the types of noise we found in our transcripts. We also do not believe that the best ASR-for-SDR measure is necessarily the one that has the highest correlation with MAP. We do follow the commonly held belief that MAP is a reasonable way of estimating IR performance, and that a poor (linear or rank) correlation with MAP indicates that a measure is targeting something different and may therefore be unsuitable as a replacement for relative MAP, regardless of its overall merit in the context of ASR for SDR evaluation.

In our experiments we recognize two distinct types of noise which result from the ASR and subsequent indexation process: transcript errors and story segmentation errors. We investigate how well the proposed methods are able to capture the impact of either of these types of noise on MAP.

4.3.1 Properties of the Test Collection

As with all of the experiments in this thesis, we used the English language portion of the TDT-2 broadcast news speech corpus, see Section 2.2.4. We perform standard IR tasks on this collection using the 99 queries that were developed by NIST for the TREC8 and TREC9 SDR benchmarks. This section provides an overview of the properties of this collection and our metadata that are important for the experiments we perform.

Transcripts The audio in the TDT-2 corpus was collected in 1998 and 1999. We have eight different automatic transcripts, seven of which were generated in 2000 in the original benchmark context, and one more recently in 2008. The latest transcript was done by CNRS-Limsi specifically for our experiments and represents a current state-of-the-art system for high-speed English language BN-speech transcription. There is also a full manual reference transcript available, but this is not up to typical benchmark standards as it consists mostly of closed captions and therefore may include rephrasings of the speech content. In addition, there are 10 hours of high quality LDC[1] transcripts.

The WER of the closed captions was estimated as ranging from 7.5 to 14.5% for radio and television sourced material respectively [Garofolo et al., 2000b], and our best estimate of the WER of the ASR transcripts ranges between 15 and 30%. We expect the 'errors' in the closed captions to be mostly deliberate re-phrasings of the true content, whereas ASR errors are typically of a more random nature. As all our measures are against the same reference we expect our measures to overestimate the true number of errors, as the total error we find is a combination of the errors in the reference and the errors in the ASR transcript. It is also likely that at some locations the ASR transcript is a more literal representation of the speech than the closed captions. We do not expect this to result in overall better retrieval performance for the ASR-based experiments though. Since the exact impact of noise in the references is difficult to establish without expending an enormous amount of effort, we shall assume that the resulting bias is similar for all transcripts and does not need to be dealt with explicitly.

Analysis of the various automatic transcripts shows that there is a high overlap in errors made by the LVCSR systems. The eight different automatic transcripts were produced by four labs/systems: Limsi (3x), Cambridge University (2x), Sheffield University (2x) and NIST (1x). Intra-lab overlap in errors was around 80%, whereas inter-lab overlap was less at around 65%. A higher overlap in errors indicates that systems make mostly the same errors, with a better system simply making less rather than different errors. A high overlap is likely to reduce the difference between quantitative and qualitative approaches to ASR-for-SDR evaluation.

Regarding the nature of the errors, more than 60% of all errors were made on the 50% least frequent terms, and 40% on the 25% least frequent terms, indicating that less frequent terms were more error-prone. Queries which rely on less frequent terms are therefore more likely to suffer from ASR transcript noise than queries that do not. The best performing systems in terms of WER (Limsi and Cambridge University) had a lower proportion of errors on their high-frequency terms than the worst performing systems (NIST and Sheffield University), indicating that the improved performance was mainly due to better recognition of the more frequent terms.

Before indexation by our IR system, all transcripts were normalized using

[1] http://www.ldc.upenn.edu

'tranfilt' and the 'en20010117_hub5.glm' rules, both available from NIST[2] and previously used in the context of TREC SDR. This normalization includes mapping spelling variations to a unified form, splitting certain compound words, and expanding some commonly used abbreviations.

Baseline experiments using $bm25$ on the reference transcript and using reference story boundaries showed that the TREC8 queries resulted in a higher MAP than the TREC9 queries, at .4601 versus .3432. The combined set of all queries resulted in an overall MAP of .4011. Topic/query terms showed an error rate that was significantly lower than average. On the Limsi2008 transcript, TER for query-terms was 27% whereas all non-stopwords had a TER of 38%. We therefore conclude that query terms were relatively 'easy' terms for the Limsi ASR system to transcribe.

Story Segmentations For the experiments on relative performance of the various automatic transcripts we use the reference segmentations which were used in the TREC8 SDR benchmark. The placement of story boundaries is typically found to be rather subjective, but assuming a ground truth can be established, it may be sub-optimal for use in an IR task [Cieri et al., 1999]. In the context of a Topic Detection and Tracking benchmark, boundaries were intrinsically evaluated, so performance in an IR context was not a target. If story segmentation were evaluated in an extrinsic manner, using MAP, we may find that intrinsically well-performing segmentations do not produce good results in this context. This is illustrated in Chapter 3.

When IR results based on reference boundaries are used as a ground truth, which is the case in our experiments in this chapter, suboptimal boundaries may become problematic as boundaries that differ from the reference could in theory result in improved IR performance. However, we only measure differences in performance and implicitly assume that any difference is the result of a performance reduction. Given that the segmentation alternatives that we use in our experiments in this chapter all resulted in significantly lower MAP than the reference segmentation, see Chapter 3, we expect this issue to not affect our conclusions and is therefore not further addressed.

We use story segmentations that were generated for our experiments in Chapter 3. Fixed duration segmentation is used for nine different durations, both with and without overlap, and TextTiling for twenty different configurations. Each segmentation is applied to the reference transcript and an IR task is performed using the 99 TREC8 and TREC9 SDR queries. For each of the three query sets (TREC8, TREC9, and both) the linear and rank correlation between the values of the ASR-for-SDR measures and MAP is calculated for all segmentations. A high correlation indicates that the quality of story segmentations in an SDR context can be established without using qrels.

[2] http://www.nist.gov

4.3.2 Evaluation

Both transcript noise and story segmentations lend themselves for evaluation using the extrinsic measures: Kendall's τ, τ_{AP}, Spearman's ρ, Blest's ω, Average Overlap, and Rank-Biased Overlap. Intrinsic evaluation of transcript noise is best done using TER, IER, and RIA, whereas segmentation cost is more suitable for intrinsically measuring the quality of story boundaries.

The segmentation cost (C_{seg}) is calculated using Equation 4.13. A moving window of 15 seconds is applied to a text and is scored for missed and inserted boundaries, resulting in p_{miss} and p_{fa}. The remaining parameters are set at 1 for the cost of a miss (C_{miss}) and the cost of a false-alarm (C_{fa}), and 0.1 for p_{seg}, the a priori likelihood of finding a segment boundary.

$$C_{seg} = C_{miss} \times p_{miss} \times p_{seg} + C_{fa} \times p_{fa} \times (1 - p_{seg}) \tag{4.13}$$

The experiments are performed using a custom-built search engine which uses a $bm25$ ranking function. No query or document expansion techniques are used, resulting in a somewhat basic absolute level of IR performance, and making our MAPs significantly lower than what was achieved by the participants in the TREC SDR benchmarks. We use 'terse' queries which do not contain any stopwords. For some experiments we do not use the reference story boundaries, however, all qrels are defined for the reference stories. In those cases we produce a position in the collection, which, in order to enable the normal evaluation method, is replaced with its corresponding reference story by the 'UIDmatch.pl' script. This procedure was also used in TREC SDR. The value of MAP is calculated with the 'trec_eval 9.0' program [Garofolo et al., 1997].

To determine the potential for the measures as an alternative for MAP as ASR-for-SDR measure, we calculate the rank and linear correlation coefficients between each of the alternative measures and MAP. Traditionally MAP is believed to be a reliable indicator of relative system performance, hence a primary goal for any new measure is to do the same. We used τ and ρ to establish how similarly systems were ranked as compared to MAP, and used Pearson's r to establish the linear correlation between our measures and relative MAP. A high linear correlation indicates that the ASR-for-SDR measures are suitable for predicting expected relative MAP under these conditions. The limit of significance is always at $p<0.05$, and all reported correlations are significant, unless stated otherwise. Conclusions are primarily based on retrieval experiments using all 99 queries, unless stated otherwise.

4.4 Results

4.4.1 Transcript Noise

Table 4.1 shows the MAPs that were achieved on each of the transcripts of the TDT-2 collection. Clearly, the TREC8 queries are much easier than the TREC9 queries, and MAP values of different transcripts are sometimes very

	TREC8	TREC9	All
Reference	0.4601	0.3432	0.4011
Limsi-2008	0.4310	0.3343	0.3822
Limsi-1u	0.4134	0.3179	0.3652
Limsi-2u	0.4211	0.3230	0.3716
CU-htk-s1u	0.4199	0.3221	0.3705
CU-htk-s1p1u	0.4045	0.3069	0.3552
Nist-b1u	0.3960	0.3167	0.3559
Shef-1k	0.3937	0.2975	0.3451
Shef-2k	0.4056	0.2986	0.3516

Table 4.1: *MAP of baseline retrieval experiments using 'TREC8', 'TREC9', or 'All' query sets for nine different transcripts of the TDT-2 collection.*

close, making transcript ranking based on MAP rather precarious. The highest MAP was achieved using the reference transcript, followed by the newest Limsi transcript, both as expected. Third in rank is the Limsi-2u transcript, followed by CU-htk-s1u and Limsi-1u. The relative performance of Nist-b1u and Shef-2k depends on which queries are used. We find that there is perfect rank correlation between MAP resulting from just the TREC9 queries and the combination of TREC8 and TREC9 queries for these transcripts.

Intrinsic Evaluation For each transcript and each query set we calculated (stopped and stemmed) TER, (stopped and stemmed) IER, and (stopped and stemmed) RIA. Table 4.2 shows rank correlation between IR results on ASR transcripts and the reference, using Kendall's τ and Spearman's ρ, indicating the ability of the three measures to correctly rank the transcripts for their expected MAP. The 'TREC8' and 'TREC9' columns show TER, IER, and RIA used in an extrinsic manner, as they are calculated on only query terms. The values in the 'All' column are for TER, IER, and RIA when used intrinsically, so when calculated on the full collection.

The ranking based on TREC8 queries is the hardest to predict for TER, IER, and RIA, perhaps because this ranking also deviates from the ranking found using all 99 queries. Using stopped and stemmed data results in an improvement for IER and RIA on the TREC9 queries. If we assume that using a higher number of queries results in a better ranking of transcript quality in an SDR context, then the 'All' column contains the results that are of the most interest. When ranking transcripts based on all 99 queries, and calculating TER, IER, and RIA using all terms in the collection, we find perfect rank correlation between MAP and the three intrinsic approaches.

Table 4.3 is similar to Table 4.2, but gives Pearson's linear correlation coefficient (r), rather than rank correlations. The relatively high correlations indicate that TER, IER, and RIA provide an accurate prediction of relative MAP. Linear correlation is highest when MAP is calculated using all 99 queries. Stemming

	TREC8		TREC9		All	
	τ	ρ	τ	ρ	τ	ρ
TER	0.7857	0.9048	0.9286	0.9762	1.0000	1.0000
IER	0.7857	0.9048	0.9286	0.9762	1.0000	1.0000
RIA	0.7857	0.9048	0.9286	0.9762	1.0000	1.0000
Stopped and Stemmed						
TER_{st}	0.7857	0.9048	0.9286	0.9762	1.0000	1.0000
IER_{st}	0.7857	0.9048	1.0000	1.0000	1.0000	1.0000
RIA_{st}	0.7857	0.9048	1.0000	1.0000	1.0000	1.0000

Table 4.2: *Rank correlation of TER, IER, and RIA with MAP. The 'TREC8' and 'TREC9' columns indicate only queries from those query sets were used for evaluation, the 'All' columns are based on the full collection.*

	TREC8	TREC9	All
TER	0.9077	0.9215	0.9891
IER	0.8933	0.9642	0.9939
RIA	0.9130	0.9580	0.9951
Stopped and Stemmed			
TER_{st}	0.9097	0.9311	0.9915
IER_{st}	0.9029	0.9617	0.9916
RIA_{st}	0.9132	0.9620	0.9980

Table 4.3: *Linear correlation of TER, IER, and RIA with MAP. The TREC8, TREC9, and All columns indicate which terms were used for evaluation.*

and stopping results in a slightly (not significantly) higher linear correlation for TER and RIA, but not for IER (except for the TREC8 queries). Of these methods, RIA_{st} has the highest linear correlation with MAP at 0.9980. We have only eight transcripts to compare, resulting in relatively large margins of error on the correlations. As a result, none of the methods can be shown to have a significantly higher correlation with MAP than any of the others for any query set.

Extrinsic Evaluation We used the six extrinsic methods that were discussed in Section 4.2, with three values for the bias control parameter of RBO. Tables 4.4 and 4.5 contain the rank and linear correlation of the extrinsic ASR-for-SDR measures and MAP-based evaluation.

Table 4.4 shows that Blest's ω and Average Overlap result in perfect rank correlation with MAP for the TREC9 queries and the full set of queries, and have the same rank correlation as IER and RIA for the TREC8 queries. The other methods typically show one additional swap, or one less in the case of

	TREC8		TREC9		All	
	τ	ρ	τ	ρ	τ	ρ
Kendall's τ	0.7143	0.8810	0.9286	0.9762	0.9286	0.9762
τ_{AP}	0.7143	0.8810	0.9286	0.9762	0.9286	0.9762
Spearman's ρ	0.7143	0.8810	0.9286	0.9762	0.9286	0.9762
Blest's ω	0.7857	0.9048	1.0000	1.0000	1.0000	1.0000
Average Overlap	0.7857	0.9048	1.0000	1.0000	1.0000	1.0000
$RBO_{0.98}$	0.7857	0.9048	0.9286	0.9762	0.9286	0.9762
$RBO_{0.95}$	0.8571	0.9286	0.9286	0.9762	0.9286	0.9762
$RBO_{0.90}$	0.8571	0.9286	0.9286	0.9762	0.9286	0.9762

Table 4.4: *Rank correlation of extrinsic ASR-for-SDR measures with MAP.*

	TREC8	TREC9	All
Kendall's τ	0.8621	0.9632	0.9461
τ_{AP}	0.8550	0.9595	0.9413
Spearman's ρ	0.8098	0.9380	0.9014
Blest's ω	0.8617	0.9447	0.9342
Average Overlap	0.9172	0.9650	0.9808
$RBO_{0.98}$	0.9514	0.9554	0.9939
$RBO_{0.95}$	0.9632	0.9481	0.9965
$RBO_{0.90}$	0.9717	0.9229	0.9911

Table 4.5: *Linear correlation of extrinsic ASR-for-SDR measures with MAP.*

$RBO_{0.95}$ and $RBO_{0.90}$. Given how close MAPs actually are for these 9 transcripts, we don't think these small differences in rank correlation are sufficient for drawing any conclusions on the usefulness of these methods in an ASR-for-SDR evaluation workflow.

In Table 4.5 we find that the linear correlation between overlap-based methods and MAP seems higher than for the correlation-based methods, but the differences in correlation in this table are only significant for $RBO_{0.95}$ and $RBO_{0.90}$ versus τ, τ_{AP}, ρ, and ω, and for $RBO_{0.90}$ versus ρ. The overlap-based methods are roughly at the same level as the intrinsic methods, whereas the correlation-based methods are worse (but correlation is still highly significant). Of the extrinsic evaluation measures, $RBO_{0.95}$ is the most promising for transcript noise, as it has a high rank correlation and the best linear correlation with MAP of the tested methods.

4.4.2 Story Segmentation

Fixed Length We first focus on the fixed-length segmentation from Section 3.2.1. Segments with nine different durations using non-overlapping windows were created on a reference transcript of the TDT-2 collection and the IR tasks were run on each segmentation. Table 4.6 and Table 4.7 show the rank and

	TREC8		TREC9		All	
	τ	ρ	τ	ρ	τ	ρ
Kendall's τ	0.6667	0.7833	0.6111	0.7333	0.5000	0.6333
τ_{AP}	0.6667	0.7833	0.5556	0.6667	0.5556	0.7000
Spearman's ρ	0.6111	0.7667	0.7222	0.8667	0.5000	0.6333
Blest's ω	0.6667	0.7833	0.5000	0.6000	0.5556	0.7000
Average Overlap	0.6667	0.7833	0.5000	0.6000	0.5556	0.7000
$RBO_{0.98}$	0.8333	0.9333	0.5556	0.6667	0.7222	0.8000
$RBO_{0.95}$	0.4444	0.6000	0.7222	0.8500	0.7778	0.8667
$RBO_{0.90}$	0.4444	0.6000	0.7778	0.9000	0.6667	0.8000
C_{seg}					0.6111	0.7500

Table 4.6: *Fixed-length/NO: Rank correlation of MAP with extrinsic ASR-for-SDR evaluation measures.*

	TREC8	TREC9	All
Kendall's τ	0.9469	0.8963	0.9661
τ_{AP}	0.9538	0.9024	0.9694
Spearman's ρ	0.8967	0.9104	0.9579
Blest's ω	0.9634	0.8894	0.9713
Average Overlap	0.9709	0.8999	0.9783
$RBO_{0.98}$	0.9639	0.9126	0.9818
$RBO_{0.95}$	0.9517	0.9246	0.9816
$RBO_{0.90}$	0.9309	0.9324	0.9775
C_{seg}			0.9508

Table 4.7: *Fixed-length/NO: Linear correlation of MAP with extrinsic ASR-for-SDR evaluation measures.*

linear correlation with MAP for the extrinsic evaluation measures using rank correlation and overlap-based methods. In addition, C_{seg} is shown for comparison purposes as this is the intrinsic method that is traditionally used for segmentation evaluation.

A comparison between Tables 4.6 and 4.4 shows that ranking systems based on expected retrieval performance as a result of segment length, is not as successful as for ASR transcript noise. For the full set of 99 queries, $RBO_{0.95}$ has the highest rank correlation with MAP, although it is lower than any of the rank correlations for transcript noise, and is also the lowest of all the methods for the TREC8 queries (but a bias setting of 0.98 gives it the highest rank correlation for those queries). The segmentation cost has a rank correlation that is similar to the extrinsic methods.

Despite the somewhat lower rank correlation for fixed duration segmentation, linear correlation is rather similar to what was found for transcript noise, see Table 4.7. The differences between Tables 4.5 and 4.7 are not significant for any of the methods. All extrinsic methods outperform C_{seg} for linear correla-

	TREC8	TREC9	All
Kendall's τ	0.9619	0.9364	0.9700
τ_{AP}	0.9709	0.9292	0.9673
Spearman's ρ	0.9226	0.9282	0.9671
Blest's ω	0.9808	0.9373	0.9692
Average Overlap	0.9842	0.9420	0.9739
$RBO_{0.98}$	0.9844	0.9376	0.9738
$RBO_{0.95}$	0.9870	0.9358	0.9724
$RBO_{0.90}$	0.9872	0.9361	0.9696
C_{seg}			0.6674

Table 4.8: *Fixed-length/O: Linear correlation of MAP with extrinsic ASR-for-SDR evaluation measures.*

tion with MAP, but not significantly. Although none of the tested evaluation methods is significantly different from any of the others, for fixed-length non-overlapping segmentation, the highest linear correlation with MAP is achieved using $RBO_{0.98}$.

In Chapter 3, the non-overlapping windows approach showed an anomaly on the TREC9 queries for the 60 and 70 second segment duration where MAP dropped unexpectedly (see Figure 3.5). We therefore implemented fixed-length segmentation with windows that overlapped by 50%, removing this anomaly without substantially altering retrieval performance otherwise. Table 4.8 is similar to Table 4.7, except segmentation with overlapping windows was used.

The use of overlapping segments practically disqualifies the cost function as an evaluation method, for obvious reasons. Linear correlation for the TREC9 queries is somewhat higher than for non-overlapping segments for all other measures, but not significantly. The difference is smallest for the RBO-based methods. The most likely explanation is that RBO has more bias towards the top of the result list, so the anomaly may have affected this measure the least for the non-overlapping segmentations. The highest correlation for fixed-length segmentation with overlapping windows is found for Average Overlap and $RBO_{0.98}$.

TextTiling Using TextTiling story segmentation, we evaluated the retrieval performance for twenty settings of two parameters (see Section 3.2.2). Table 4.9 gives the rank correlation results for the extrinsic ASR-for-SDR evaluation methods on the twenty segmentations that were generated using TextTiling. The cost function clearly has the lowest correlation with MAP (though p-values are less then 0.001), Average Overlap has the highest correlation for the TREC8 queries, and τ_{AP} has the highest rank correlation for the TREC9 queries and the full set of queries.

The linear correlation of MAP with the extrinsic ASR-for-SDR measures for the twenty TextTiling settings is shown in Table 4.10. τ_{AP} and $RBO_{0.95}$ have the highest linear correlation with MAP, although none of the extrinsic measures is significantly better than any of the others. The cost function scores

	TREC8		TREC9		All	
	τ	ρ	τ	ρ	τ	ρ
Kendall's τ	0.7619	0.8870	0.7158	0.8977	0.8000	0.9414
τ_{AP}	0.7937	0.9204	0.7474	0.9053	0.8316	0.9564
Spearman's ρ	0.7335	0.8725	0.7053	0.8902	0.7684	0.9143
Blest's ω	0.7513	0.8738	0.7158	0.8962	0.8105	0.9429
Average Overlap	0.8085	0.9188	0.7018	0.8612	0.8000	0.9414
$RBO_{0.98}$	0.7230	0.8893	0.6737	0.8406	0.7895	0.9368
$RBO_{0.95}$	0.6913	0.9022	0.6526	0.8301	0.8105	0.9489
$RBO_{0.90}$	0.6596	0.8634	0.5684	0.7654	0.8000	0.9459
C_{seg}					0.6174	0.7988

Table 4.9: *TextTiling: Rank correlation of MAP with extrinsic ASR-for-SDR evaluation measures.*

	TREC8	TREC9	All
Kendall's τ	0.9138	0.8649	0.9251
τ_{AP}	0.9398	0.8650	0.9324
Spearman's ρ	0.8898	0.8562	0.9199
Blest's ω	0.8984	0.8680	0.9162
Average Overlap	0.9439	0.8403	0.9241
$RBO_{0.98}$	0.8793	0.8057	0.9239
$RBO_{0.95}$	0.8695	0.7819	0.9331
$RBO_{0.90}$	0.8424	0.7624	0.9234
C_{seg}			0.7704

Table 4.10: *TextTiling: Linear correlation of MAP with extrinsic ASR-for-SDR evaluation measures.*

significantly worse than all of the extrinsic measures, except for ω.

4.5 Conclusion

In this chapter we investigated an alternative approach to evaluation of ASR transcripts in an SDR context. We extrinsically evaluated ASR transcripts by making a direct comparison between ranked results from IR on a reference and an automatic transcript. The primary goal is to do extrinsic evaluation without using qrels, whose outcome is highly correlated with traditional MAP-type evaluations. We tested correlation with MAP for three intrinsic methods of evaluation for transcript noise, and for six extrinsic methods for transcript noise and automatic segmentation.

For the ASR transcripts of the TDT-2 collection that we used in our experiments, a very high linear correlation with MAP was found for the baseline intrinsic measure TER, which itself is very similar to WER. This may cast doubt on whether extrinsic evaluation has any added value in the context of ASR sys-

tem optimization. However, it is important to note that the eight automatic transcripts we used were all generated by similarly operating ASR systems, of which the parameters were all optimized for the same target: WER. If one then compares systems based on that target, it is no surprise that a better system may also be better on criteria not included in the original target, as there is no reason to expect improvements for any of the systems on non-TER related aspects of performance. The main reason for differences in WER between the ASR transcripts is not due to differences in parameter settings, which can be assumed to have been optimal for each individual system, but rather due to inherently better or more (time) efficient implementations of training and decoding algorithms. Our approach to ASR evaluation allows for optimization procedures that specifically target IR performance and was shown to have as high a correlation with MAP as TER, even in conditions where the latter was the optimization criterion for the various ASR transcripts.

The outcomes of the experiments in this chapter showed that our approach is clearly feasible, as all methods we investigated had very high correlation with MAP, both for transcript noise and for automatic segmentation. The fact that our methods were shown to have as high a linear correlation with MAP as TER when used to evaluate ASR transcripts that were explicitly optimized for their TER performance, is a testament to the robustness of our extrinsic measures under various conditions. The much higher correlation with MAP than C_{seg} for automatic story segmentation further shows the value of extrinsic evaluation in an IR context.

4.5.1 Research Questions

Can we evaluate ASR transcripts in an intrinsic manner that is more appropriate for SDR than traditional WER?
Answer: Using TER, IER, and RIA we can achieve very high linear correlation with MAP of more than 0.99. All of these methods take more of the transcript-processing chain of an SDR system into account than WER, thereby removing certain irrelevant contributions to the error rate. The best correlation with MAP was found for RIA when calculated on a stopped and stemmed version of the (indexed) transcript, but the difference with TER was not significant.

Which method for extrinsic evaluation has the highest correlation with relative MAP?
Answer: All of the methods we tested in this chapter had significant rank- and linear correlations with MAP. The highest linear correlation with MAP was consistently achieved using RBO at either the 0.95 or 0.98 setting for rank bias. RBO was also consistently among the highest rank correlations, making this the best method in our testing.

Can extrinsic evaluation of ASR transcripts without qrels be used to predict MAP-based performance?
Answer: Yes. We tested this for traditional ASR transcript noise and for ar-

tificial story boundaries. In both cases we found that linear correlation with MAP was highly significant: for transcript noise, values of more than 0.99 were recorded, and for story boundaries correlations of between 0.92 and 0.98 could be obtained.

4.5.2 Summary

Extrinsic evaluation using the approach that was outlined in Section 1.3.2 requires a direct comparison between the results of two IR experiments. In this chapter we have investigated the correlation with MAP for several novel and existing approaches to comparing ranked lists. Our experiments indicate that extrinsic evaluation of ASR transcripts using our method can be as good as intrinsic evaluation for the collection and the conditions that we analyzed. The potential advantages of extrinsic evaluation in the manner that we propose include the ability to measure the quality of a system for a certain type of information need, but without the need for qrels.

When it comes to automatic story segmentation, the intrinsic evaluation method of using a cost function was shown to be suboptimal. Our proposed extrinsic measures all provided a much better approximation of the rank or value of the relative MAP and as a result are potentially better targets for optimization of an automatic story segmentation algorithm for use in a retrieval task.

When evaluating an automatic transcript and/or segmentation, all of the extrinsic methods we investigated provided significant correlations with MAP. Overall, the overlap-based methods were the most robust and RBO has the additional advantage over AO of being 'tunable' for expected user behavior. For example, if one expects a user to inspect only a few results, one could set the RBO parameter a bit lower, giving more weight to errors at the top of the result list. A bias parameter value of 0.95 was found to result in the highest correlation with MAP in our experiments.

5

Artificial Queries and Transcript Duration

In the previous chapter we investigated several intrinsic and extrinsic evaluation measures for the quality of an ASR transcript in the context of spoken document retrieval. These included intrinsic measures Term Error Rate, Indicator Error Rate, and Relevance-based Index Accuracy, and extrinsic measures based on rank correlation: Kendall's τ, Spearman's ρ, Blest's ω, and τ_{AP}, or overlap: Average Overlap and Rank-Biased Overlap. We determined the linear and rank correlation of these measures with (relative) MAP, where the ground truth was based on retrieval on a full reference transcript of the 400 hour English language portion of the TDT-2 BN collection. In the context of real-life SDR applications, it is unlikely that a 100+ hour transcript is available, making the procedure described in Chapter 4 unrealistic for most out-of-the-lab applications. Usually there is little more than the raw audio and a limited number of hours of manual transcripts available as a basis for evaluation and system optimization.

In order to make the ASR-for-SDR measures practicable, one must be able to calculate them on the basis of resources that can be realistically produced. We therefore need to know the minimum amount of manually generated resources needed for obtaining a reliable estimate of ASR-for-SDR quality. For TER, IER, and RIA, only reference transcripts are needed, but for the correlation and overlap-based methods there is also a dependence on queries. In this chapter we investigate the practical limitations for these evaluations by determining how the duration of reference transcripts affects the reliability of the performance measures. Our strategy for doing this includes measuring performance using various amounts of reference transcripts and by using differently sized sets of appropriate queries.

In order to test the impact of the amount of reference transcripts on correlation with MAP, we repeat some of our experiments from Chapter 4 but with varying amounts of references, and with stability of correlation as performance target, rather than correlation itself. Subsets of arbitrary duration can be formed by (randomly) selecting stories from our large TDT-2 collection. However, performing retrieval experiments on these subsets is not as straightforward, due to the fact that there are no queries or qrels defined for such ad-hoc

selections taken from the collection. Although it is feasible to manually define appropriate queries (and, if needed, matching qrels) for a small collection, in the context of our experiments we need to investigate *many* different sets. Manually defining queries for hundreds of subsets of TDT-2 is an unrealistic proposition. Although query generation traditionally requires an expert who is familiar with the contents of the collection, a practical alternative for our needs may be found in automatic generation of collection-specific queries. In this chapter we compare methods of automatic query generation and we investigate how the use of artificial queries affects the correlation between ASR-for-SDR evaluation and relative MAP. We address the following research questions:

- How many (artificial) queries are needed to reliably estimate ASR-for-SDR performance?
- Which method for automatic query generation results in the highest correlation between ASR-for-SDR measures and MAP as calculated from real queries?
- How is the reliability of the ASR-for-SDR performance measures affected by the duration of the manually transcribed references?

This chapter is organized as follows: in Section 5.1 we examine an existing method of automatic query generation for known-item retrieval tasks and propose how this method can be adapted for generating queries for traditional IR tasks. In Section 5.2 we shortly address the issues surrounding durational requirements on reference transcripts. Artificial queries are then used in a number of experiments that are described in Section 5.3. The results of the experiments are provided in Section 5.4, followed by the conclusions of this chapter in Section 5.5.

5.1 Automatic Query Generation

Section 5.1.1 provides an overview of an approach to automatic query generation that was developed for the task of known-item retrieval. This is followed by an explanation of how we adapted this method for query generation in our extrinsic evaluation framework in Section 5.1.2.

5.1.1 Previous Work on Artificial Queries

In [Azzopardi and de Rijke, 2006] an automatic query generation algorithm was introduced that is aimed at supporting evaluations of known-item retrieval tasks, see Section 2.2.5. Such a task can be thought of as retrieval with only one target story for each query, with the specific aim of finding this target within a larger collection of stories. Users are expected to have knowledge of the collection and some recollection of a certain story they want to find. This is an attractive scenario for a researcher, because when a query is specifically

created for a selected item there is no need for additional qrels. The focus of research was automatic query generation with an aim to produce similar values for Mean Reciprocal Rank (see Section 2.2.5) as human-generated queries. The same target stories were used for both real and artificial queries.

In a follow-up paper [Azzopardi et al., 2007], a general query generation model was introduced in which first a target story was selected from a collection, then the number of terms in the query, and finally individual query terms according to a distribution as specified in a term selection model. The general term selection model that was used is shown in Equation 5.1. The likelihood of term t_i being selected based on the properties of story d_k and user querying model m is a weighted sum of the likelihoods of the term in the target story and in the collection as a whole. The better the recollection of the target story the higher the contribution of terms from the target story, so the lower the value of λ.

$$p(t_i|\theta_m^{d_k}) = (1-\lambda) \cdot p(t_i|d_k) + \lambda \cdot p(t_i) \tag{5.1}$$

The sampling strategy employed by a user is represented by $p(t_i|d_k)$, and three different models for this were proposed in [Azzopardi et al., 2007]: a popular, random, and discriminative term selection model. The Popular Selection Model (PSM), Equation 5.2, is based on term frequency, where the assumption is that a term is more likely to be selected if it occurs more frequently in a story. The Random Selection Model (RSM), Equation 5.3, assumes an equal likelihood for any term present in a story to be used as a term in a query. The Discriminative Selection Model (DSM), Equation 5.4, assumes that terms are chosen to be discriminative from terms in other stories, in other words: selecting terms that are expected to provide the highest contrast between the 'known item' and the rest of the collection. In these equations, $n(t,d)$ is the count of term t in story d, and $b(t,d)$ represents the binary presence of a term (1 if present, 0 otherwise) in the story.

$$p(t_i|d_k)_{PSM} = \frac{n(t_i,d_k)}{\sum_{t_j \epsilon d_k} n(t_j,d_k)} \tag{5.2}$$

$$p(t_i|d_k)_{RSM} = \frac{b(t_i,d_k)}{\sum_{t_j \epsilon d_k} b(t_j,d_k)} \tag{5.3}$$

$$p(t_i|d_k)_{DSM} = \frac{b(t_i,d_k)}{p(t_i) \cdot \sum_{t_j \epsilon d_k} \frac{b(t_j,d_k)}{p(t_j)}} \tag{5.4}$$

These models were tested on a known-item retrieval task on the EuroGov corpus for six languages. All three models were found to produce queries which were significantly different from manually generated queries for Dutch, Hungarian, and German. The PSM was the best approach on English language queries,

the RSM on Spanish queries and DSM was the best for Portuguese queries. In these cases, and for the PSM on Spanish, the artificial queries did not result in a significantly different MRR from manual queries. Furthermore it was determined that the similarity in MRR between the artificial and manual queries did not depend on the collection size, nor on the size of the vocabulary.

5.1.2 Artificial Queries for Extrinsic ASR Evaluation

The ASR-for-SDR measures that were investigated in Chapter 4 were used in a traditional IR evaluation scenario that is rather different from the one in [Azzopardi et al., 2007]. Adapting their method for MAP instead of MRR seems like a daunting task, as getting similar performance for a single item was shown to be rather difficult. Without such a clear target, automatic query generation for traditional TREC-style retrieval is expected to be even harder. There is also little previous work for such a task and there seems to be much less of a need for artificial queries in traditional IR, as evaluation is more likely to be limited by the resources needed for generating qrels than by queries. However, in contrast to the work that was summarized in Section 5.1.1, we do not need our artificial and real queries to result in similar values for MRR or MAP. Instead, we focus on the behavior of our ASR-for-SDR measures and the correlation with relative MAP. Given these differences we propose an approach to query generation which is inspired by the three term-selection models from [Azzopardi et al., 2007].

For a collection with human-made queries and qrels, we re-interpret each query as multiple known-item retrieval tasks. For each of these tasks, one of the stories that were judged as relevant is designated as the known-item. In the case of the TDT-2 collection for example, there are ~ 4000 relevant stories for 99 queries. This gives us ~ 4000 known-item retrieval tasks. We collect the following statistics on these tasks:

- The terms in each query are divided into two categories for each target story: in-target-story terms and out-of-target-story terms. The distribution in number of terms in each of these two groups is recorded.
- For the group of in-target-story terms, we record the distribution of $p(t_i|d_k)$ based on the assumption of having been selected using PSM, RSM, and DSM.
- For the group of out-of-target-story terms (which do not occur in the target story), we record the distribution of $p(t)$.

As the queries we use were not originally conceived as known-item queries, they contain relatively many terms that are not-in-target-story. For these terms $p(t_i|d_k)$ is always 0. We therefore handle these separately from the in-story terms and characterize them by their $p(t)$. Due to the fact that the number of out-of-target-story terms was (much) higher than the in-target-story terms for the TREC-SDR queries we used for analysis, we generate only half as many out-of-target-story terms as we found in the TREC-SDR queries.

Using the statistics that were collected in this manner from the real queries from TREC-SDR, we generate new queries according to any of the three term selection models and the following procedure:

- Choose a term selection model to use for calculating $p(t_i|d_k)$.
- Select a target story d_k from the collection. The length of d_k must be at least $x\%$ of the average length of all stories in the collection.
- Generate a query length, with the number of in-story terms and out-of-story terms randomly determined following the recorded distributions from real queries.
- For each in-story term, generate a target $p(t_i|d_k)$ according to the recorded distributions from the real queries and relevant stories.
- For each generated $p(t_i|d_k)$, randomly select a term from the $y\%$ of terms in the story that are closest to this target. Add this term to the query.
- For each out-of-story term, generate a target $p(t)$ according to the recorded distributions from the real queries and non-relevant stories.
- For each generated $p(t)$, randomly select a term from the $y\%$ of terms in the collection that are closest to this target, and are not in d_k. Add this term to the query.

Some speech collections contain a relatively high number of particularly short stories. For example, the TDT-2 English BN speech collection has an average story duration of 173 words, with half of the stories less than 86 words long. In a BN context short stories may not have much content, but could be filler content or overviews of upcoming stories. We want to avoid having these non-content stories as retrieval units, as they are unlikely to be targeted by real queries. A higher value of x results in the selection of longer targets, thereby increasing the likelihood of the target being a content story.

The selection of terms cannot be too rigid, as this may limit the number of possible queries for a given target story. When the collection contains only those stories that were manually transcribed, as is the case in ASR-for-SDR evaluation, the number of stories is likely to be limited. However, given that artificial queries are relatively easy to generate, we may want to use relatively many of them. This means that we may need to generate multiple queries for a single target. With an average of 102 unique terms for each story in the TDT-2 collection, there is a risk that the term selection models always converge towards the same terms. To ensure that the selection mechanism does not result in the same terms being selected repeatedly, we allow for a certain amount of deviation from the target statistical properties for each term. The allowed amount of deviation can be controlled using y, with a high value of y resulting in the target $p(t_i|d_k)$ or $p(t)$ being only loosely followed, and a lower value resulting in a 'tighter' selection of terms.

5.2 Amount of Reference Transcripts

Traditionally, ASR evaluations using WER are performed on a small subset of the full task. Provided the subset is representative of the collection, the resulting WER should give a reliable estimate of the number of errors that can be expected in a full transcript. Since a human-made transcript was available for the entire English part of the TDT-2 collection, we used it in Chapter 4 as ground truth to get the best possible gauge of ASR-quality. For WER, the validity of the subset approach has been shown [Young and Chase, 1998], but for our IR-based ASR evaluations it is unclear how the size of a randomly selected subset affects the correlation of ASR-for-SDR evaluation with MAP.

For most evaluations, it is assumed that subsets of a collection can be chosen in such a way that they are representative of overall (or rather average) quality, even if there is much variation in the speech. For WER, this can be tricky as an ASR system may for example have a WER of <10% on clean studio speech, whereas spontaneous conversational speech is transcribed with a WER that exceeds 50%. In a collection that contains a mix of both, it is important that they are each represented in the subset and in the right amount. Without detailed knowledge of the entire collection, this may mean that a large subset must be used in order to ensure it is representative of the full collection.

For our extrinsic ASR-for-SDR measures, which are based on IR performance, it is unclear how one should go about selecting stories from the collection for inclusion in an evaluation. Although ASR transcript quality is expected to be a factor in such an evaluation, IR evaluation is content-based, so content may also be needed as a selection criterion. To avoid such issues, and because we do not wish to assume any knowledge of the collection, we assume that the selection of a 'representative' subset is done in a random manner. To obtain a subset of a given length, multiple fragments from random locations in the collection are included. The main restriction is that fragments must be typical retrieval units, usually 'stories'. For an unsegmented collection this can be implemented by randomly generating positions in the collection and then manually finding the closest natural boundaries around this location. In the case of a collection for which (full) reference story boundaries are available, these may be selected directly. One can add randomly selected stories until the duration requirement is met.

5.3 Experimental Setup

The methods that were proposed in Chapter 4 for ASR-for-SDR evaluation were found to have a high rank and linear correlation with the relative MAP of systems based on transcripts with varying amounts of noise. As was done in TREC8 and TREC9 SDR, we assume that the impact of transcript noise on IR is reflected in relative MAP. Although the selection of stories for the evaluation and the choice of queries are likely to have an impact on the *absolute* values of our ASR-for-SDR evaluation measures, their *relative* values are expected to

remain similar across various automatic transcripts, at least for queries that conform to broadly similar characteristics. This is important as we cannot expect to have a full reference transcript, and must be able to estimate relative retrieval performance on a selection of stories for which 'real' queries are not available. For measuring transcript noise using the proposed extrinsic measures, we need a reference transcript of an appropriately sized subset of the collection, and queries that are suitable for doing a traditional IR task on this selection of stories.

With our experiments we want to determine: i. how many (artificial) queries are needed for ASR-for-SDR evaluation, ii. which method for automatic query generation results in the highest correlation between our measures and MAP, and iii. how the reliability of our measures is affected by the duration of the reference transcript. The second one is done by comparing correlation between MAP on real queries and the ASR-for-SDR measures on real and artificial queries generated using each of the three query term selection methods. The first and third are achieved by testing various sets of a given amount and analyzing how much the choice of queries/stories impacts the correlation between MAP and the ASR-for-SDR measures. If we assume that on average any selection of queries/stories results in the correct values of ASR-for-SDR measures, and therefore a high correlation with MAP, then the standard deviation can be used to determine how likely it is for any particular selection to be sufficiently representative of the collection to ensure this high correlation. We expect that the larger the amount of queries/transcripts, the lower the standard deviation of the ASR-for-SDR values and more importantly, the lower the standard deviation of the correlations.

In Section 5.3.1 we describe the experiments for determining the required number of queries for ASR-for-SDR evaluation, followed by our approach to determining the best way of generating artificial queries in Section 5.3.2. How we investigate the relationship between the amount of transcripts and the reliability of the extrinsic measures is explained in Section 5.3.3, and finally Section 5.3.4 contains a description of the test collection and the configuration of the IR system we use.

5.3.1 Number of Queries

The choice of queries may determine the absolute value of the ASR-for-SDR measures, but this need not impact their rank or linear correlation with relative MAP. See for example the queries of TREC8 and TREC9 SDR, which resulted in *absolute* MAPs of 0.46 and 0.34 respectively on the reference transcript, but whose *relative* values were rather similar over nine different transcripts. Absolute values of the evaluation measures are therefore not our main concern, as these are mainly determined by the difficulty of the queries from an ASR perspective. For example, queries which contain OOV terms are likely to result in larger differences between reference and ASR transcript performance than queries that contain typically 'easy' terms, such as the TREC8 and TREC9 SDR queries (see Section 4.3.1). However, if there is much variation in the absolute values of the ASR-for-SDR measures as a function of the queries that

we artificially generate, this may indicate that individual queries have a large impact on the outcome of the experiments. Because we do not typically control these queries individually, this is something we would prefer to avoid.

Our aim in this experiment is to determine how stable our measures are as a function of the number of queries that are used in the evaluation. When calculating MAP, 50 queries are used as a minimum to get a stable ranking of systems, although many more queries may be needed to stabilize the absolute value of MAP. It is however important to note that the 50 real queries that are used in benchmarking conditions are all carefully selected, so the presence of particularly hard queries is often quite intentional. In the case of artificial queries, the presence of 'difficult' queries must be a function of the query generation algorithm, and not depend on random factors. As our proposed measures require no qrels, we may generate many more queries than is typically done for MAP evaluations, thereby ensuring a more uniform result.

The stability of our measures can be expressed using the relative standard deviation (%RSD), which is the standard deviation as a percentage of the average value from several runs. Our procedure is as follows:

- Generate a large set of queries.
- Randomly select n sets of m queries from this set and perform an IR experiment using these queries on the full collection.
- Calculate the values for each of the extrinsic measures based on the results.
- Repeat this t times and take the means of the t average values and standard deviations of the measures.

The resulting mean average standard deviation can then be expressed as a percentage of the mean average value, giving us the %RSD. For our experiments we used $n = 100$, $t = 10$, and varied m between 10 and 500 queries, with a total pool of 2500 queries.

5.3.2 Artificial Queries

The three query generation models that were explained in Section 5.1.1 can be used to generate new queries following the patterns of other – typically human generated – queries, as explained in Section 5.1.2. Each of these three models is based on a different assumption of the human query generation process. We aim to determine which model results in queries that best mimic the behavior of real queries in the context of the extrinsic evaluation measures, and whether this behavior is sufficiently similar for artificial queries to be used in ASR-for-SDR evaluation.

We distinguish two dimensions for the quality of artificial queries: How similar are they to real queries for ASR-for-SDR? and How useful are they in the context of predicting relative MAP and ranking of transcripts with varying amounts of noise? The correlation between the results for the various measures

using real and artificial queries is used to determine their similarity. The correlation with MAP as estimated using 99 real queries and corresponding qrels is the basis for determining the usefulness of the artificial queries in the context of ASR-for-SDR evaluation.

Three sets of n (chosen based on the results of the experiments described in Section 5.3.1) artificial queries are generated using the query generation models and the methodology as outlined in Section 5.1.2. We use $x = 50$ and $y = 20$ and the TDT-2 collection and TREC8 and TREC9 SDR queries for learning the distributions of the query terms. An IR experiment is performed on eight different noisy ASR transcripts and a human-generated reference transcript for the three sets of 250 artificial and the 99 real queries. MAP is calculated for the real queries using corresponding qrels, for all nine transcripts. Eight different extrinsic measures: Kendall's τ, τ_{AP}, Spearman's ρ, Blest's ω, Average Overlap, $\text{RBO}_{0.98}$, $\text{RBO}_{0.95}$, and $\text{RBO}_{0.90}$ are used to calculate the similarity between ranked results on the reference and the noisy ASR transcripts for each set of artificial queries. We then calculate correlation with MAP, which is based only on real queries(!). We also compare the values of the extrinsic measures between real and artificial queries directly.

A high rank correlation with MAP indicates that the artificial queries result in values for the measures that can be used to rank noisy transcripts in accordance with how they would be ranked if real queries with corresponding qrels were used. High linear correlation means that the measures can be used to estimate the relative value of MAP for an automatic transcript, which is a measure for the impact of transcript noise on IR performance. A high rank or linear correlation between the values of the measures on the real and the artificial queries indicates that the artificial queries behave similarly to the real queries in the context of evaluating transcript noise.

Using these correlations we compare the performance of the output of three automatic query term selection models to the real queries to see which model is the best fit for our our purposes and whether its performance is sufficient for the use of artificial queries in the experiments that must be done to determine the amount of transcripts needed for ASR-for-SDR evaluation.

5.3.3 Amount of Transcripts

We test how much transcribed speech is needed from the TDT-2 collection, assuming a random selection of stories, to have a reasonable chance of it being representative of the full collection, in the context of both the intrinsic and extrinsic evaluation measures from Chapter 4.
Our procedure is as follows:

- Choose 15 durations to test.
- For each duration, select 10 sets of stories from the collection that have a total length that is equal to this target.
- For each set of stories ($10 \times 15 = 150$), generate n artificial queries.

- Perform an IR experiment on eight automatic and the reference transcript for each of the $n \times 150$ sets of artificial queries (and story sets). This results in $9 \times 150 = 1350$ IR tasks with n queries each.

For each of the 15 durations we then calculate the %RSD in the values of measures, as well as the rank and linear correlations between MAP on the full collection and the values of the intrinsic and extrinsic ASR-for-SDR measures of the evaluation sets. We choose n based on the results of the experiments described in Section 5.3.1.

A lower %RSD for in the values of the measures means that there is an increased likelihood that a random choice of stories of that duration is representative for the collection as a whole. We expect that more (longer) transcripts result in a lower %RSD for our measures. Our goal is to determine how much the %RSD depends on the duration of the speech transcripts in the context of the various measures.

We distinguish three contexts for the use of ASR-for-SDR measures:

- As an absolute indication of transcript quality.
- As an indication of relative performance of an IR system that uses the transcript.
- As a means to estimate relative MAP.

In the first of these contexts, the relationship between the duration of transcripts and %RSD of the raw measures needs to be examined, for the second and third context, the %RSD of the rank and linear correlation respectively between the measures and MAP is of interest.

5.3.4 Test Collection and IR Configuration

As with our experiments in Chapter 4 we used the English language subset of the TDT-2 Broadcast News speech collection collected in 1998 and 1999 for performing IR experiments, see Section 2.2.4 . For this collection we have 99 topics (queries) with corresponding qrels which were developed by NIST for the TREC8 and TREC9 SDR benchmarks[Garofolo et al., 2000b] in 2000 and 2001. We used eight different automatic transcripts, seven of which were generated in 2000 in the original benchmark context, and one more recently in 2008. The latest transcript was done by CNRS-Limsi specifically for our experiments and represents a current state-of-the-art system for high-speed English language BN-speech transcription.

Additionally we used a full manual reference transcript of the 400 hours of speech. A small part of this reference (around 10 hours) was done to benchmark standards and provided by LDC[1]. The remainder was made up of closed-caption quality transcripts with a WER that was estimated as ranging from 7.5 to 14.5%

[1] http://www.ldc.upenn.edu

# of queries	10	25	50	100	250	500
Kendall's τ	6.20	4.13	2.86	1.99	1.23	0.80
τ_{AP}	6.33	4.02	2.96	1.99	1.23	0.81
Spearman's ρ	5.29	3.28	2.26	1.66	1.01	0.65
Blest's ω	5.25	3.20	2.28	1.62	0.96	0.64
Average Overlap	4.54	2.86	1.95	1.36	0.83	0.58
$RBO_{0.98}$	6.85	4.26	2.95	2.10	1.31	0.85
$RBO_{0.95}$	7.73	4.82	3.39	2.36	1.44	0.96
$RBO_{0.90}$	8.86	5.53	4.06	2.73	1.72	1.13

Table 5.1: *%RSD for eight ASR-for-SDR evaluation measures for various amounts of queries.*

for radio and television sourced material respectively [Garofolo et al., 2000b]. This introduces an additional source of noise, as our reference may contain errors which are not present in the automatic transcripts. Although the difference between reference transcript and what was really said is likely to have some impact on the values of the ASR-for-SDR measures, we cannot easily determine the amount of impact, but will assume it is small enough to not influence our conclusions.

All IR experiments were done using a custom IR system which used the $bm25$ ranking function with $b = 0.5$ and $k_1 = 1.1$. Up to 1000 results were generated for each query. The limit of significance is always at $p<0.05$, and all reported correlations are significant, unless stated otherwise.

5.4 Results

5.4.1 Number of Queries

Using the PSM algorithm, see Section 5.1.1, we generated 2500 queries for the (complete) English part of the TDT-2 BN speech collection following the method described in Section 5.1.2. IR experiments were done on both the reference and the Limsi automatic transcript from 2008. We then selected queries from this set of 2500 using the method described in Section 5.3.1, with $n = 100$, $t = 10$ and with m at 12 values between 10 and 500. For each selection of queries we calculated the value of the eight ASR-for-SDR correlation and overlap-based evaluation measures.

Figure 5.1 shows the relative standard deviation as a function of the number of queries that was used to determine the value of the measures: Kendall's τ, τ_{AP}, Spearman's ρ, Blest's ω, Average Overlap, $RBO_{0.98}$, $RBO_{0.95}$, and $RBO_{0.90}$. All of the measures follow the same pattern: more queries results in lower %RSD, and therefore more stable values. Table 5.1 contains the exact values for m equal to 10, 25, 50, 100, 250, and 500.

Some methods show an intrinsically lower %RSD than others, but this does not mean they can be used with fewer queries than the other measures. It simply

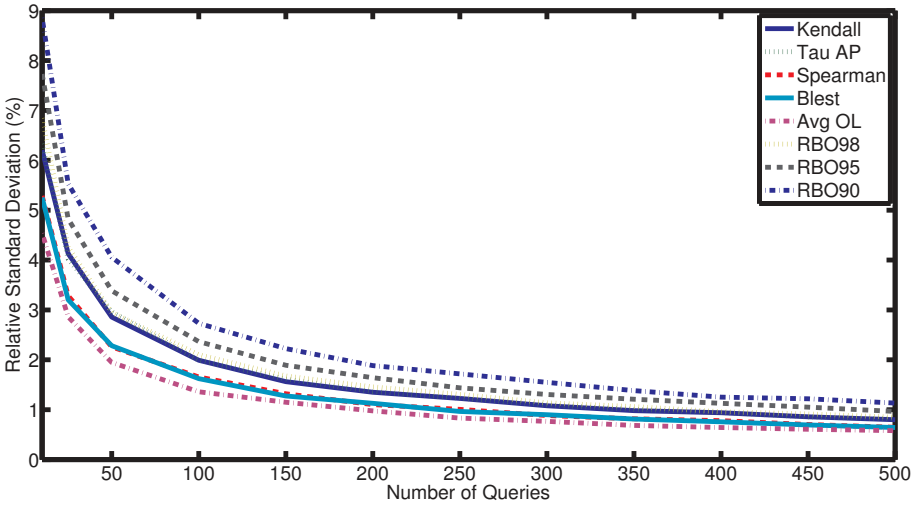

Figure 5.1: *%RSD as a function of the number of queries used for determining the value of eight extrinsic ASR-for-SDR evaluation measures.*

reflects the fact that some methods naturally show less variation in their absolute values. For example, Kendall's τ and Spearman's ρ are highly correlated and have the same range of -1 to +1, but the value of Spearman's ρ varies (much) less in these conditions. The Overlap-based methods have a range between 0 and 1, but only have a value of 0 when two sets are disjoint rather than simply having uncorrelated rankings. So although it may seem as if Average Overlap requires roughly half the number of queries as compared to Kendall's τ to achieve the same stability, this need not have any consequences for the rank and/or linear correlation of these values with MAP.

The main message to take away from Figure 5.1 is that all measures react in a similar manner to an increase in the number of queries, and that more queries results in more stable values for our measures which holds true up to the limits of our experiments at 500 queries. One needs about quadruple the amount of queries to achieve a 50% reduction in %RSD.

5.4.2 Artificial Queries

We compared rank and linear correlation between the 'real' TREC8 and TREC9 SDR queries and 250 artificial queries, generated using three different user models on the same collection. The results are shown in Tables 5.2, 5.3, and 5.4, for the user models based on the popular, random, and discriminative selection models respectively. These tables show the rank correlation using Kendall's τ and Spearman's ρ, and linear correlation using Pearson's r between the 250 artificial queries and the 99 real queries. The 'MAP' column shows the rank and

linear correlations between relative MAP as calculated from the 99 real queries and the values of the various measures as calculated on the 250 artificial queries. The 'real' columns give the correlations between the values of the various measures as calculated on the 99 real queries and the 250 artificial queries using the same measure.

If we assume that MAP as achieved on the real queries provides a good measure of the quality of the ASR in an SDR context, then a high correlation in the 'MAP' columns implies that our artificial queries are suitable for estimating the ASR quality. A high correlation in the 'real' columns implies that our measures behave similarly on the artificial queries as they do on the real queries for various amounts of transcript noise. As the results in Tables 5.2, 5.3, and 5.4 show, all three models result in significant and high rank and linear correlations with real queries.

A comparison between the three query generation models in their similarity to the 99 real queries (using the 'real' columns), shows that DSM typically scores lower than the other two on rank and lower than RSM on linear correlation, except for Kendall's τ on rank and linear correlation and Spearman's ρ on linear correlation. PSM and RSM are similar on rank correlation, except for $\text{RBO}_{0.98}$ and $\text{RBO}_{0.95}$. On linear correlation, RSM scores slightly higher on all methods except for $\text{RBO}_{0.98}$. This leaves the overall impression that RSM results in queries that behave most similarly to real queries on these measures. PSM is worst in terms of linear correlation, whereas DSM is worst for rank correlation. None of the differences between the various methods and query generation models are significant though.

Comparing the three query generation models on their usefulness in a context of ASR-for-SDR evaluation (using the 'MAP' columns), again shows that DSM scores lower than the other two models on rank correlations, except for $\text{RBO}_{0.95}$ and $\text{RBO}_{0.90}$, but in this case also on linear correlation. PSM and RSM are again very similar on rank correlation, with RSM only better on $\text{RBO}_{0.95}$. For linear correlation PSM is better for each method except for Average Overlap, $\text{RBO}_{0.95}$, and $\text{RBO}_{0.90}$ where both systems are virtually tied. The best query generation model for use in an ASR for SDR evaluation context therefore seems to be PSM, although the difference with RSM is very small.

For all the different evaluation methods, a high rank and linear correlation is found between real and artificial queries, with results for $\text{RBO}_{0.95}$ and $\text{RBO}_{0.90}$ slightly lower than the rest. The use of artificial queries instead of human generated queries has no appreciable impact on the usefulness of these evaluation methods, in terms of linear or rank correlation with MAP. The best query term selection model from the perspective of predicting system ranking and relative MAP is PSM, which gives both the highest linear and rank correlation for almost all evaluation methods.

5.4.3 Amount of Transcripts

We recognize three aspects of the stability of the various measures as a function of the amount of transcripts used: i. the values of the measures, ii. the stability

	τ		ρ		r	
	MAP	real	MAP	real	MAP	real
Kendall's τ	1.000	0.929	1.000	0.976	0.996	0.961
τ_{AP}	0.929	1.000	0.976	1.000	0.995	0.959
Spearman's ρ	0.929	1.000	0.976	1.000	0.995	0.924
Blest's ω	0.929	0.929	0.976	0.976	0.995	0.949
Avg Overlap	1.000	1.000	1.000	1.000	0.996	0.977
$RBO_{0.98}$	0.929	1.000	0.976	1.000	0.989	0.973
$RBO_{0.95}$	0.714	0.786	0.881	0.905	0.980	0.976
$RBO_{0.90}$	0.786	0.857	0.905	0.929	0.974	0.975

Table 5.2: *PSM: Correlations between MAP on 99 real queries and ASR-for-SDR evaluation measures on 250 artificial queries.*

	τ		ρ		r	
	MAP	real	MAP	real	MAP	real
Kendall's τ	1.000	0.929	1.000	0.976	0.992	0.977
τ_{AP}	0.929	1.000	0.976	1.000	0.991	0.974
Spearman's ρ	0.929	1.000	0.976	1.000	0.990	0.948
Blest's ω	0.929	0.929	0.976	0.976	0.989	0.969
Avg Overlap	1.000	1.000	1.000	1.000	0.997	0.982
$RBO_{0.98}$	0.929	0.857	0.976	0.952	0.985	0.969
$RBO_{0.95}$	0.857	0.929	0.952	0.976	0.980	0.977
$RBO_{0.90}$	0.786	0.857	0.905	0.952	0.976	0.981

Table 5.3: *RSM: Correlations between MAP on 99 real queries and ASR-for-SDR evaluation measures on 250 artificial queries.*

	τ		ρ		r	
	MAP	real	MAP	real	MAP	real
Kendall's τ	0.857	0.929	0.952	0.976	0.983	0.981
τ_{AP}	0.857	0.929	0.952	0.976	0.988	0.972
Spearman's ρ	0.857	0.929	0.952	0.976	0.970	0.962
Blest's ω	0.857	0.857	0.952	0.952	0.980	0.967
Avg Overlap	0.857	0.857	0.952	0.952	0.985	0.944
$RBO_{0.98}$	0.786	0.714	0.905	0.881	0.965	0.936
$RBO_{0.95}$	0.857	0.786	0.952	0.905	0.965	0.957
$RBO_{0.90}$	0.857	0.786	0.952	0.905	0.963	0.959

Table 5.4: *DSM: Correlations between MAP on 99 real queries and ASR-for-SDR evaluation measures on 250 artificial queries.*

of system ranking as they follow from the values of the measures, and iii. the linear correlation of the measures with relative MAP as achieved using real queries on the full collection.

Experiments were performed on subsets of several different durations that were taken from the full collection. We evaluated for durations of 30, 60, 120, 180, 240, 300, 360, 420, 480, 540, 600, 750, 900, 1050, and 1200 minutes. For each duration, we randomly selected stories from the collection until the required duration was met. We then generated queries using PSM, specifically for the selected stories. For the shorter durations of 30 and 60 minutes 100 queries ($n = 100$) were generated, for all other durations we used 250 queries ($n = 250$). The stability of the measures when used with artificial queries was established in Section 5.4.1. For 100 queries this was between 1.4 and 2.7%RSD (depending on the measure used), and for 250 queries it was between 0.8 and 1.7%RSD. Each retrieval experiment was repeated with ten different subsets of each length, on nine different transcripts.

Stability of the raw values Figure 5.2 shows the %RSD of the eight extrinsic methods of ASR for SDR evaluation, versus the duration of the manual transcripts used. As expected, the measures become more stable as more references are used. As there is a linear relation between the amount of effort required and the amount of transcripts we have, it is important to know how much is actually needed in order to get a fair assessment of performance. For all of the eight measures shown in Figure 5.2, around 180 minutes of reference transcripts results in less than $\sim$ 3%RSD. Increasing the duration of the reference transcripts beyond 180 minutes gives slightly more stable results, but clearly provides diminishing returns.

The results for the intrinsic evaluation measures are shown in Figure 5.3. The 't' at the end of the name of the measure indicates that its values was calculated using only query terms, making it an extrinsic evaluation. TER and IER show very similar %RSD values, which remain above 4% until eight hours of transcripts were used, and slowly improve with more transcripts. A much lower %RSD is found for RIAt, with a value of only 2% when using 30 minutes of references and remaining below 1% for most other (longer) durations.

As with Figure 5.1 the %RSD of the individual measures is of little overall importance as the main aim is to obtain high correlation with MAP. No hard conclusions can be drawn from the %RSD of the absolute values.

Stability of linear correlation The %RSD's of the measures are not our main concern, as they do not say much about the stability of linear and rank correlation, which is what is needed when using them to determine the quality of a transcript in an SDR context. Linear correlation for the extrinsic methods is shown in Figure 5.4, and for the intrinsic methods in Figure 5.5.

A comparison between Figures 5.2 and 5.4 confirms that the absolute value of %RSD for the measure says little about the %RSD of its linear correlation with MAP. Average Overlap had the most stable raw value, but when used to

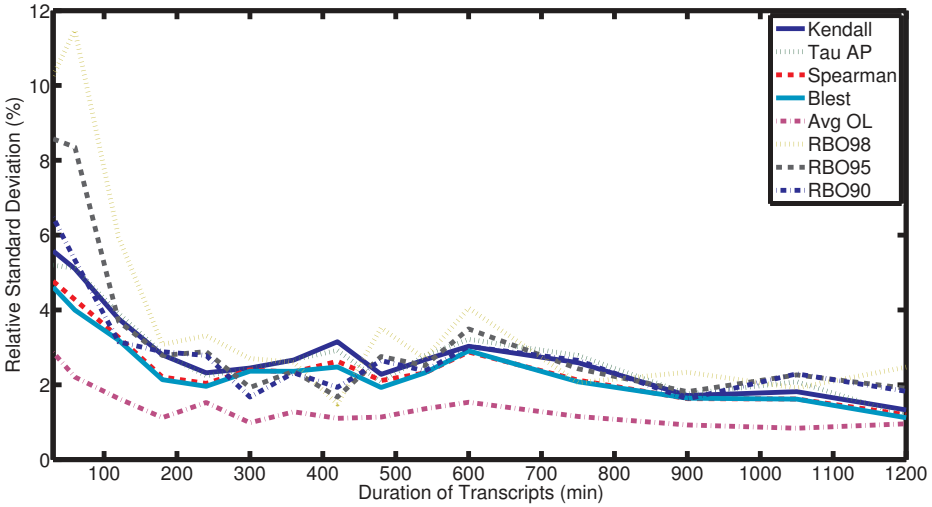

Figure 5.2: *%RSD of the extrinsic ASR-for-SDR evaluation measures as a function of the amount of transcripts used to determine their value.*

calculate linear correlation it is not significantly different from the other extrinsic measures. It is however the only measure to consistently not exceed 2% RSD for each transcript duration of more than 180 minutes. This 180 minutes duration seems to be the sweet spot for most measures, except for $\text{RBO}_{0.98}$ which still has an %RSD of over 6% at this point.

The peaks in Figure 5.4 at 300 and 750 minutes for Blest's ω and Spearman's ρ may be caused by the (random) choice of stories for the evaluation. We used 'only' ten runs for each duration, so they may have been caused by one or two unfortunate selections of stories. What is interesting however, is that the same stories (and queries) did not cause issues for the other measures, possibly indicating that ω and ρ result in a linear correlation that is intrinsically less stable in the face of story selection than the other measures.

A comparison between Figures 5.3 and 5.5 shows that the lower duration stability of TER and IER as compared to RIA has no impact on the duration stability of the linear correlation with relative MAP. In fact, although the 't' versions of the measures are roughly equal in Figure 5.3 to their non-'t' counterpart, they show a consistently better performance in the duration stability of the linear correlation. For the intrinsic measures, a duration of around 180 minutes of manual transcripts seems to be a reasonable target for obtaining a reliable estimate of relative MAP.

Stability of rank correlation Similar to the previous paragraph, the %RSD of the Kendall's τ rank correlation between the values of the extrinsic and intrinsic measures and relative MAP are shown in Figures 5.6 and 5.7. Notice

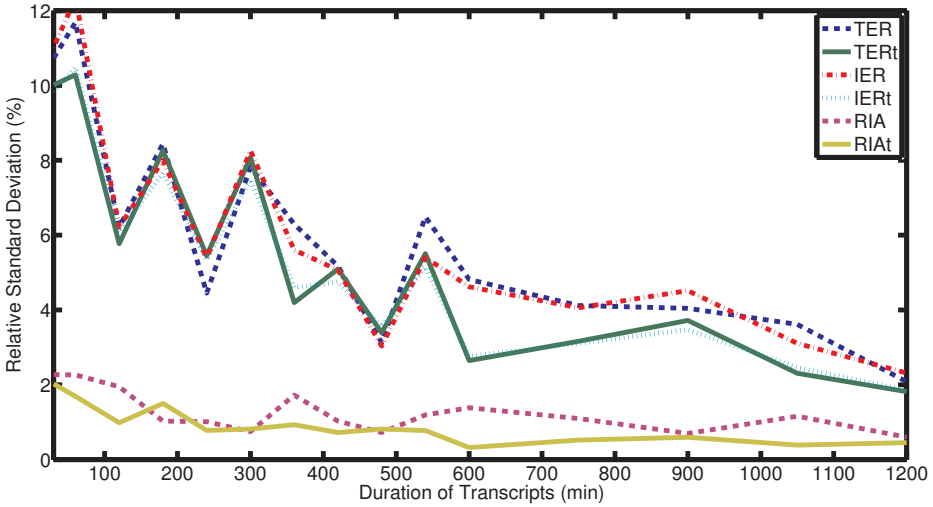

Figure 5.3: *%RSD of the intrinsic ASR-for-SDR evaluation measures as a function of the amount of transcripts used to determine their value.*

the vertical scale which is larger than in the previous figures to accommodate for the fact that %RSD is generally larger for rank correlation than for linear correlation when there are only eight transcripts to rank. Most of the measures hover between 5 and 10% RSD, with Average Overlap and Kendall's τ the only extrinsic measures to remain below 7% for all durations of more than 180 minutes. The intrinsic measures show very similar behavior, except for RIAt whose system ranking is more stable in the face of changing queries/stories than the other measures. For the extrinsic measures and the topic-specific intrinsic measures, 240 minutes seems to be a reasonable minimum amount of transcripts to use for evaluation of system ranking.

5.5 Conclusion

We set out to determine how many (artificial) queries are required for ASR-for-SDR evaluation using extrinsic measures, which method of automatic query generation results in the highest correlation with MAP, and how the reliability of the evaluation is affected by the amount of reference transcripts.

Our experiments have shown that increasing the number of queries results in a more stable, and therefore more reliable, estimate of ASR quality using the extrinsic measures. Assuming a (rather arbitrary) target of ~ 3%RSD, one needs at least 100 artificial queries. Furthermore, no appreciable impact on rank or linear correlation between the extrinsic measures and MAP was found due to the use of artificial queries on the TDT-2 collection. In fact, linear correlation

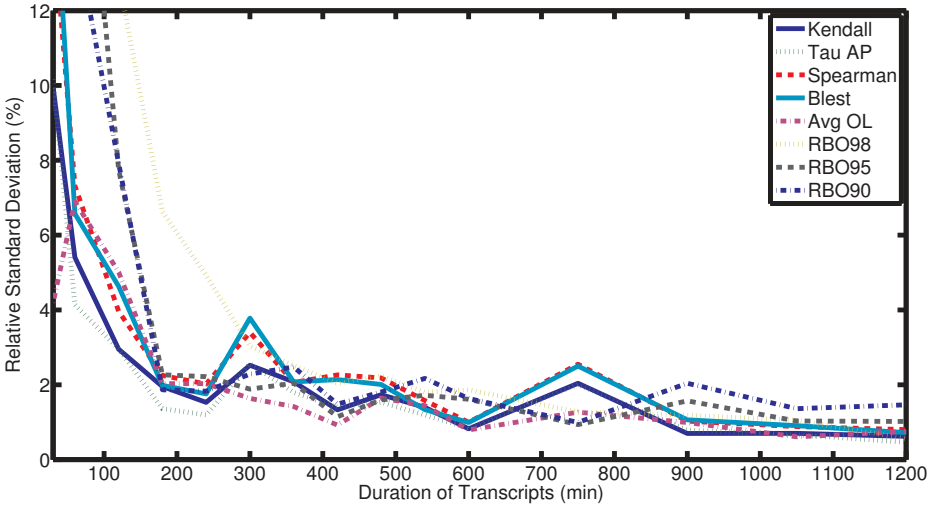

Figure 5.4: *%RSD of linear correlation with MAP for the extrinsic ASR-for-SDR evaluation measures as a function of the amount of transcripts used to determine their value.*

between ASR-for-SDR measures on 250 artificial queries and MAP on 99 real queries was slightly higher than when both were calculated on the same 99 real queries for most measures.

The amount of transcripts that is needed for extrinsic evaluation is somewhat dependent on the requirements of the application. To obtain a reliable linear correlation, 3 hours was shown to be enough for less than ∼3%RSD. Rank correlation on eight transcripts results in a somewhat higher %RSD for almost all reference transcript durations, but using at least 4 hours of manual transcripts seems to be a reasonable compromise between performance and effort.

These findings show that the ASR-for-SDR measures that were investigated in Chapter 4 are not only viable from a performance point of view, but can also be implemented without a need for large amounts of resources. Although the manual transcripts must be selected in such a way that they cover complete 'stories', there is no need for more material than would be required for traditional ASR evaluation using WER. For a somewhat 'generic' approach to extrinsic evaluation, one could generate artificial queries and use these to get an initial impression of ASR-for-SDR performance. A real in-depth analysis that specifically targets a usage scenario requires queries that are representative of such a scenario, but there is no need for generating 'expensive' qrels, making this a rather feasible proposition for most collections.

We conclude that extrinsic evaluation of ASR quality in an SDR context can be done reliably without needing qrels and without excessive amounts of reference transcripts. Given the availability of off-the-shelf IR solutions, extrinsic

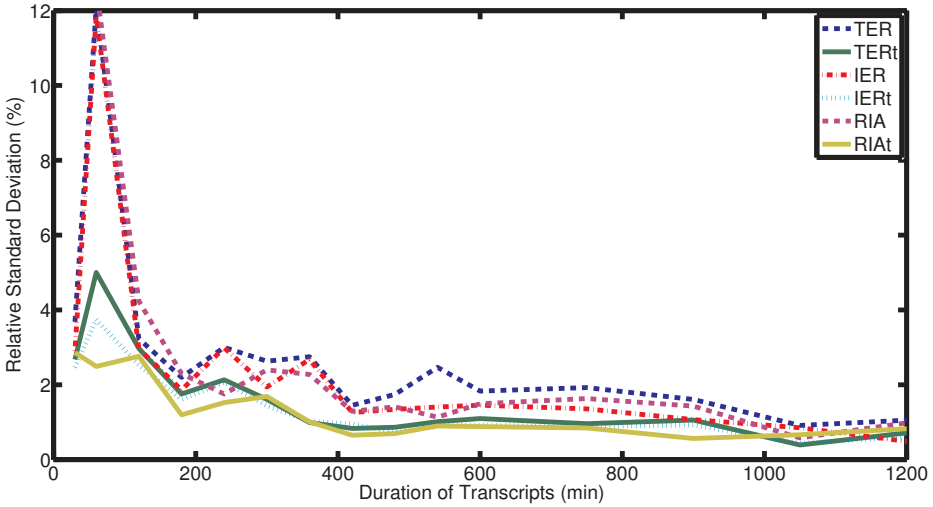

Figure 5.5: *%RSD of linear correlation with MAP for the intrinsic ASR-for-SDR evaluation measures as a function of the amount of transcripts used to determine their value.*

ASR-for-SDR evaluation is as accessible as traditional WER for anyone using an LVCSR system for transcribing spoken document collections, but should result in an evaluation that is better adapted to the context.

5.5.1 Research Questions

How many (artificial) queries are needed to reliably estimate ASR-for-SDR performance?
Answer: All of the extrinsic ASR-for-SDR measures we tested reacted in a similar way to an increase in the number of queries, with a quadrupling of queries resulting in half the amount of %RSD. For artificial queries it makes sense to use as many as the size of the collection and time available for evaluation supports. For human-made queries, one must be prudent, as developing queries, even without qrels, requires an amount of effort, whilst the returns are diminishing. Our experiments showed that for the collection we examined, 100 queries was a reasonable minimum, however, as there was no need for qrels and we used artificial queries, increasing the number of queries was relatively cheap. We therefore used 250 queries in most of our experiments.

Which method for automatic query generation results in the highest correlation between ASR-for-SDR measures and MAP as calculated from real queries?
Answer: We compared three automatic query generation algorithms for their similarity in behavior to real queries and their usefulness in estimating relative

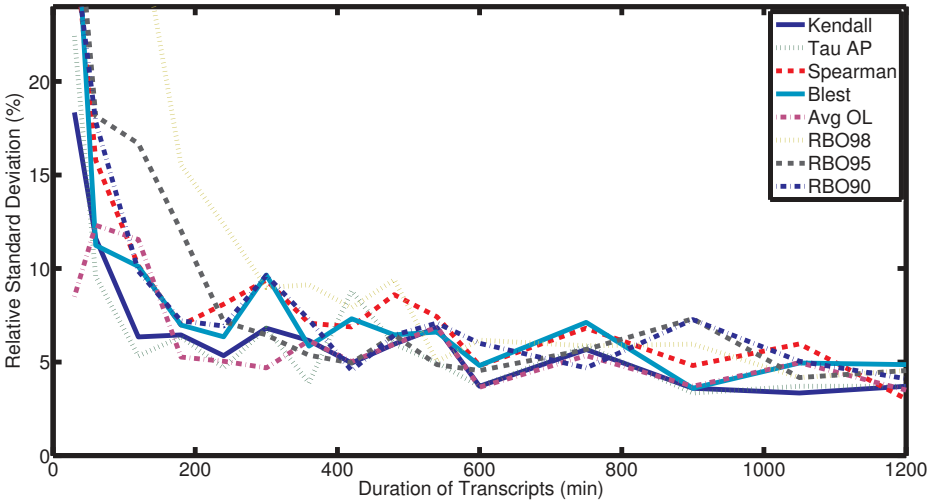

Figure 5.6: *%RSD of Kendall's rank correlation with MAP for the extrinsic ASR-for-SDR evaluation measures as a function of the amount of transcripts used to determine their value.*

MAP and relative rank of eight different noisy transcripts. All methods resulted in high correlations and are expected to yield useful results. In a direct comparison, the popular term selection model performed slightly better than the other two on linear correlation with relative MAP, though not significantly, and was therefore chosen for use in our other experiments.

How is the reliability of the ASR-for-SDR performance measures affected by the duration of the manually transcribed references?
Answer: Linear correlation between the values of the intrinsic and extrinsic measures and relative MAP stabilized at around 2% RSD when 3 hours of transcripts were used, increasing the amount of transcripts significantly (to 10+ hours) caused a further reduction to around 1% RSD in our experiments. Rank correlation between the ASR-for-SDR measures and relative MAP. stabilized from around 4 hours of reference transcripts.

5.5.2 Summary

We have shown that extrinsic evaluation of ASR transcripts using a very limited amount of manually created resources can be nearly as good as similar evaluations using a full transcript of a 400-hour speech collection and set of 99 hand-made queries. This means that extrinsic evaluation of ASR-for-SDR can be done without expending more effort than for traditional WER-based evaluations.

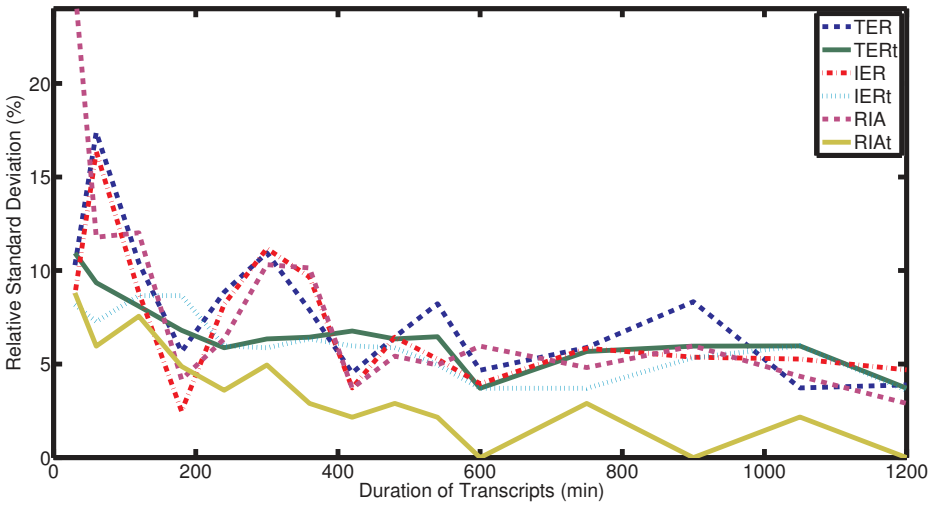

Figure 5.7: *%RSD of Kendall's rank correlation with MAP for the intrinsic ASR-for-SDR evaluation measures as a function of the amount of transcripts used to determine their value.*

If there is no possibility to manually generate representative information requests, one can automatically generate queries. The best of the methods we investigated resulted in a higher linear correlation with MAP for most of the ASR-for-SDR measures than using real queries. If there is a need to test specific information requests, for example, with a focus on named entities, one may need to employ different automatic query generation strategies. We have not tested these as there was no comparable human-made set with qrels available to test for such conditions.

Extrinsic evaluation of ASR-for-SDR can be done using artificial queries and with as little as 3 hours of manually transcribes speech. This makes this type of evaluation as accessible as WER-based evaluation, but with the distinct advantage of being better attuned to the needs of spoken document retrieval. In addition, extrinsic evaluation grants the possibility to adapt the evaluation for a specific information need through the choice of queries, or for an expected application through tuning the evaluation depth.

6

Summary and Conclusion

6.1 Summary

In this thesis we have introduced a novel framework for the evaluation of ASR transcripts in an SDR context. An overview of this framework was given in Chapter 1 and is its visualization is repeated in Figure 6.1. The basic premise is that ASR transcripts must be evaluated by measuring the impact of noise in the transcripts on the search results of a traditional IR task. In this framework, calculating ASR-for-SDR is done through a direct comparison between ranked result lists of IR tasks on a reference and a hypothesis transcript.

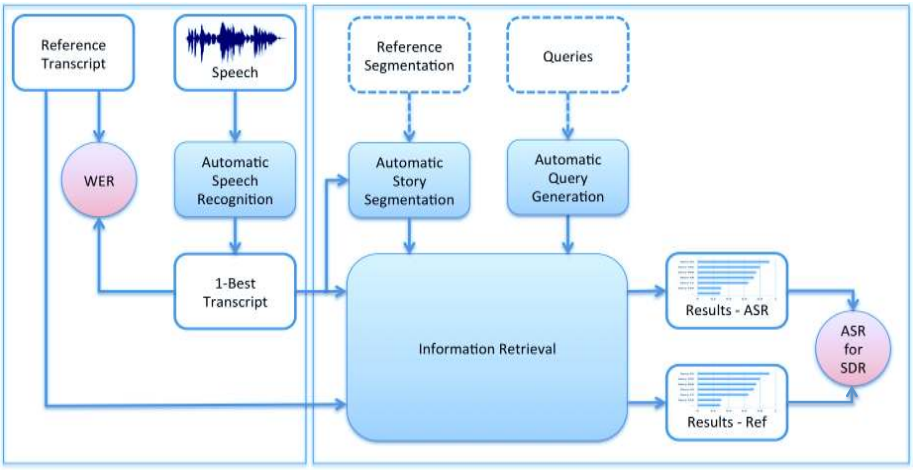

Figure 6.1: *Overview of a proposed novel framework for ASR-for-SDR evaluation without the use of qrels or a need for manually generated queries and story boundaries.*

After a short summarization of previous work in automatic speech recognition and information retrieval in Chapter 2, we started our investigation into the various aspects of this framework with a comparison of various automatic story segmentation algorithms in an SDR context, see Chapter 3. We showed that

intrinsic evaluation results in different conclusions with regard to the quality of automatic story boundaries than when MAP is used. This indicates that when automatic story segmentation is used for search applications, the traditionally used segmentation cost may not be a good performance target. The highest MAP was achieved using a dynamic segmentation method of our own design, but this method cannot be implemented directly within the context of a traditional IR system as it segments on-the-fly for each query, whereas normally pre-segmented text is indexed. Its performance may also be more dependent on query and collection properties than the other segmentation methods we investigated, so we do are hesitant to recommend this method until it has been tested under a wide range of conditions. A seemingly more robust approach is chopping up the text or speech into segments of a constant (fixed) duration. On our collection, this was functionally as good as using more advanced techniques such as TextTiling, or a WordNet-based approach. All of the approaches to automatic story segmentation resulted in a relatively large reduction in MAP of around 10-15% relative. Improvements in automatic story segmentation for SDR therefore provide an enormous potential for advancing the performance of SDR.

In Chapter 4 we proposed several methods for the ranked result comparison (ASR-for-SDR) in Figure 6.1. A key property of these methods is a lack of reliance on relevance judgments (qrels). Our premise was that if the results of these evaluations have a high linear correlation with (relative) MAP, they are an extrinsic measure for the impact of ASR transcript noise or story segmentation errors on retrieval performance. If the results have a high rank correlation with MAP, they can be used to rank various approaches or configurations for their relative expected performance when used as part of an SDR system. Linear correlation with MAP for the various ASR transcripts and segmentations was found to be highly significant, with $\mathrm{RBO}_{0.95}$ reaching 0.997 for transcript noise. Rank correlation varied between essentially perfect and highly significant. Although some methods had a slightly higher linear correlation than others, the differences were rarely significant. We therefore conclude that all methods are equally suitable for this task. The choice of which method to deploy comes down to the aspects that are expected to be most important in the use of the system, for example, the expected persistence of the user in the inspection of results, or properties of the interface.

After proving the theoretical viability of the proposed framework, we looked at practical aspects in Chapter 5. We investigated how much reference transcripts were necessary to achieve the high correlations with MAP that were found in Chapter 4. This was done by testing on a large number of subsets of various sizes from a 400-hour collection, necessitating the use of artificial queries rather than the real TREC SDR queries we had used up to that point. We developed an automatic query generation algorithm that was able to generate artificial queries that resulted in our measures having as high a correlation with MAP as when real queries are used. If we allow for a relative standard deviation of the linear correlations at a (somewhat arbitrary) maximum of 3%, we can estimate ASR-for-SDR performance using as little as three hours of reference

transcripts. This amount is roughly equal to what is required to do traditional intrinsic evaluation using WER. We therefore concluded that extrinsic evaluation of ASR-for-SDR performance can be done as easily as intrinsic evaluation, without needing more or different resources. Although we strongly recommend using human-generated queries that truly reflect the envisaged use of the SDR system, artificial queries were found to be a reasonable alternative when real queries cannot be easily obtained.

6.2 Conclusion

The benefit of the extrinsic ASR-for-SDR evaluation framework that we propose, is that it combines the desirable properties of the traditional approaches to evaluation of ASR transcripts for use in SDR, WER and MAP, while avoiding their limitations. The upside of WER is that it can be calculated using only a reference transcript for a relatively small amount of speech, but its downside is that it does not provide a differentiation between the importance of words that are affected by the errors in an SDR context. For MAP, the opposite is the case: MAP gives a good insight into the consequences of transcript noise on the user experience, but it requires a prohibitive amount of resources, in particular relevance judgments for all queries in a test set and on all stories in the collection. The aim of the framework we propose is to provide a measure for the quality of an ASR transcript in an SDR context that has a high correlation with MAP, while requiring only a relatively low amount of reference transcripts and a small set of queries as resources.

We investigated several approaches for ASR-for-SDR calculation, including correlation-based and overlap-based methods. We distinguish two potential sources of transcript noise: story segmentation and transcript errors. Both are typical for the kind of noise that distinguishes speech transcripts from traditional textual sources. Correlation with MAP is somewhat higher for transcript errors than for segmentation errors, at 0.997 for the former, and between 0.92 and 0.97 for the latter, depending on the segmentation method used. Both correlations are very high, and confirm that the output of our proposed framework can be expected to be equally useful as MAP for extrinsic transcript evaluation. For story segmentation errors, the new approach is especially useful, since traditional intrinsic evaluation is only able to achieve a linear correlation of <0.8 with MAP for most configurations. The method we found to have the highest correlation with MAP was Rank-Biased Overlap.

The amount of resources required for extrinsic evaluation was estimated by establishing the relative standard deviation of the correlation with MAP for many sets containing various amounts of reference transcripts. We compared the %RSD between intrinsic and extrinsic approaches to ASR transcript evaluation, and found no appreciable difference between the two for the stability of the outcome as a function of the amount of reference transcripts used. In other words: our proposed framework requires the same amount of reference transcripts as traditional intrinsic ASR evaluation for a given level of reliability

of correlation. We established that around 3-4 hours of reference transcripts is a reasonable minimum for a homogeneous BN collection such as we used in our experiments.

Although queries are an important part of extrinsic evaluation, there may be situations in which it is undesirable to have to generate these manually, for example when evaluating ASR performance on a collection in a language for which a native speaker is not available. We have investigated whether it is also possible to automatically generate artificial queries following patterns that were learned from human-generated queries. In our experiments, artificial queries showed similar correlations with MAP as real queries, making them a feasible alternative for situations in which real queries cannot be easily obtained. This also means that extrinsic evaluation using our proposed framework can be done using the same amount of human-generated resources as traditional intrinsic evaluation, while retaining a very high correlation with MAP.

In this thesis we have demonstrated that it is possible to combine the benefits of both traditional approaches to ASR-for-SDR evaluation, while avoiding their downsides. Extrinsic evaluation of ASR transcripts can be done using only reference transcripts as a ground truth, and queries to represent a realistic usage scenario. Some questions remain though. For example, we have only tested our approaches on one collection, the English part of the TDT-2 BN speech collection. Although we have avoided collection-specific solutions wherever possible, we cannot guarantee that our approaches are generally applicable without testing under a wide range of conditions. It is unlikely that suitable test-collections can be found for all potential circumstances, but we hope that this approach is picked up by other researchers and applied to realistic, out-of-the-lab collections. This way we hope to establish the practical limitations of our approach and make refinements where needed.

A number of applications for our framework have not been discussed in this thesis. These include evaluation of ASR lattices and lattice-indexation methods, evaluation in the context of spoken term detection, evaluation of phoneme-based search, and evaluation of ASR transcripts that are not explicitly optimized for WER. These are all applications that are not properly served by intrinsic evaluation using for example WER, and for which we expect our extrinsic framework to be a promising prospect. We would welcome the opportunity to test our approach on these applications, as we expect that our framework is more capable of recognizing their respective challenges than traditional intrinsic approaches to evaluation.

6.3 Miscellaneous Musings

In the course of this research project, we have always attempted to stay as close as possible to real-life problems and collections. All challenges we tackled were based on benchmark conditions, and not fabricated to fit our chosen solution. As such, we are happy to see the high correlations between our measures and MAP, and the relatively low amount of transcripts needed to achieve them. Some other open issues in the field of spoken document retrieval may also benefit from the use of realistic tasks, but have so far been limited to artificial testing because of the large amount of resources otherwise required. This section contains some initial thoughts on SDR-related research tracks where the evaluation method we investigated in this thesis may help to further development.

Lattices One of our aims at the start of the project was to improve SDR performance by exploiting speech recognition lattices in a novel way, more specifically by infusing information that was collected from queries into the lattice decoding process. This idea was inspired by preliminary experiments on a Dutch non-broadcast-news speech collection for which we had a WER of ∼60%, but where a best path through the recognition lattices could be found that had a mere ∼20% WER. Testing whether the better paths through the lattices would improve the system, implies measuring retrieval performance, but with no qrels, this was virtually impossible. Although it would be feasible to test some of our ideas on this Dutch set using hand-crafted queries and evaluate using precision@10 for example, this would still mean a relatively intricate evaluation procedure, with little guarantee that the results are representative of real-life usage.

Extrinsic evaluation on a benchmark task, by default using MAP, is the only way to conclusively show the benefits of any novel approach to exploiting ASR lattices for SDR. But after we spent more than a month analyzing lattices of state-of-the-art ASR on the TDT-2 collection (courtesy of Limsi), we concluded that no paths could be found that resulted in a higher MAP on the benchmark tasks than the 1-best decode. As a result, no lattice indexation or retrieval technique could possibly achieve an increase in MAP. This may simply be the logical result of optimal system design, but perhaps much of this high performance was a result of the fact that the ASR system was particularly well adapted for dealing with the kind of data that was found in the TDT-2 collection. Many interesting collections are however from a domain for which much less training material is available. As a result, 1-best transcripts may not be as good as for TDT-2, and lattices have the potential for improving performance for such collections.

The benefits of lattices in SDR may be limited to collections which are challenging for ASR, for example due to a lack of matching training material. But testing the quality of a lattice cannot be done intrinsically as lattices are typically not an end-product and desirable properties of lattices are largely application-dependent. If TREC-style evaluation is needed to conclusively show

the benefits of a lattice-based approach, but qrels for MAP calculation are typically only available for collections that are unlikely to benefit from lattices, then developing lattice-based retrieval approaches becomes an arduous task. Our proposed framework should make this much easier, as it allows for extrinsic evaluations without qrels, using transcripts that can be generated quite feasibly for most ad-hoc collections, including those that are expected to benefit from lattice-based approaches to retrieval.

Phoneme-based Retrieval Some attempts at tackling the problem of out-of-vocabulary (OOV) terms in SDR have used a phonetic (or subword-based) transcript as a basis for indexing. This frees the system of the limitations of a limited-size lexicon, but also removes important linguistic information in the process. Phoneme-based retrieval is almost guaranteed to improve performance for OOV terms, but is unlikely to benefit in-vocabulary query terms. With the OOV rate of state-of-the-art LVCSR systems often under 1%, superficially it seems there is not much potential for such an approach. The main attraction of tackling OOVs is however not in their amount, but in their expected importance in the context of SDR. Although rare terms in general are often too obscure to be used in queries, Named Entities may be attractive query terms for IR users as they are relatively unambiguous. Especially in the context of BN-search and historical collections, the inclusion of Named Entities in queries can easily help to focus a search task.

Assessing the benefits of phoneme-based search is not easy in a TREC-style setting, resulting in such approaches being potentially underappreciated. On SDR benchmark collections such as TDT-2 and TDT-3, we found no OOVs in the available test queries for our lexicon/language model. Alternative tests using artificial tasks, either through limiting the lexicon or through generating queries with OOVs, may be able to show theoretical gains, but are unconvincing for showing the benefits in real-life settings. We expect that our proposed framework may be of help here, as it can be easily applied to any collection, including those for which Named Entities can be expected to naturally occur in queries. Examples are collections that reference historical events and figures, or collections that are used in the context of genealogical research.

An interesting issue to be addressed is the required amount of transcripts needed in scenarios of use for such collections. Unlike the queries in our experiments described in Chapter 5, phonetic search targets primarily OOVs. With an expected OOV-rate of $\sim$2%, this means that we may need more transcripts to show the potential benefits of phonetic search. We would be very interested in testing current approaches to phoneme-based retrieval with our proposed framework to see what amount of transcripts is required for a proper evaluation, and if the reported positive results on artificial tasks can be confirmed using our proposed method of evaluation.

Beyond TREC-style Evaluation In Chapter 2 we explicitly mentioned that information retrieval can be an iterative process in which a user adapts a query

based on initial results. In the Cranfield approach to evaluation, this step is not included as it is would complicate the evaluation procedure no end. However, it is reasonable to expect that users adapt their behavior to their experience with the system. For Spoken Document Retrieval, a simple feedback mechanism saying something along the lines of 'Harry Potter was not found in our dictionary, could you please rephrase this part of your query', may overcome many of the problems of OOVs. Similarly, if a user is presented with confidence scoring for the transcript, this may help in rephrasing queries so as to avoid the limitations of a noisy transcript.

Our proposed framework for ASR-for-SDR evaluation allows for such scenarios. One could for example simply monitor each step in a multi-query search and see whether the result on a noisy transcript becomes more similar to that on a reference transcript as a query is refined. It is then possible to adapt the interface and feedback mechanisms of the SDR system to investigate whether a potential convergence in results can be achieved more quickly – using less iterations – when a user is provided with more information on why certain results were produced. In other words, our evaluation framework allows for evaluation of more facets of the SDR process than TREC-style evaluation and can be expected to be a very useful tool in user studies on spoken document retrieval. We would be interested to see whether users adapt their searches to the limitations of the system and if so, how this is reflected in the ASR-for-SDR scores coming form the refined queries as compared to the initial attempts.

Samenvatting

In dit proefschrift introduceren we een nieuw raamwerk voor het evalueren van automatische spraakherkenningstranscripten (ASR-transcripten) voor gebruik in zoeksystemen voor gesproken documenten (SDR). Een overzicht van dit raamwerk is gegeven in Hoofdstuk 1 en een visualisatie kan worden gevonden in Figuur 6.1. De basisgedachte is dat ASR-transcripten moeten worden geëvalueerd door de invloed te meten van ruis (herkenningsfouten) in de transcripten op de zoekresultaten van een traditionele informatie zoek (IR) opdracht, dit in tegenstelling tot de traditionele manier waarbij herkenningsfouten simpelweg geteld worden (WER). In dit raamwerk wordt de kwaliteit van ASR-in-SDR-context berekend door een rechtstreekse vergelijking tussen de geordende resultatenlijst van IR-taken op een referentie- en een herkenningstranscript.

Na een korte samenvatting van eerder werk op het gebied van ASR en IR in Hoofdstuk 2, beschrijven we de verschillende aspecten van dit raamwerk aan de hand van een vergelijking tussen verschillende algoritmes voor het automatisch opdelen van documenten (tekst of spraak) in inhoudelijk bij elkaar horende segmenten voor gebruik in SDR, zie Hoofdstuk 3. We hebben laten zien dat intrinsieke evaluatie leidt tot andere conclusies over de kwaliteit van de gevonden segmentgrenzen dan wanneer we de IR-evaluatiemaat MAP gebruiken. Dit geeft aan dat als automatisch opdelen van documenten wordt gebruikt voor zoektoepassingen, de traditioneel gebruikte evaluatie met behulp van een segmentatiekostenberekening mogelijk niet tot optimale resultaten leidt. Van de door ons onderzochte methoden, werd de hoogste MAP behaald met behulp van een door onszelf voorgestelde dynamische opdeling. Deze methode kan echter niet direct worden geïmplementeerd als onderdeel van een traditioneel IR-systeem, aangezien het documenten ter plekke opdeelt voor elke zoekvraag. Normaal gesproken maken IR systemen gebruik van voorgesegmenteerde tekst om te indexeren. De prestaties van de dynamische segmentatieaanpak zijn mogelijk ook meer afhankelijk van de zoekvraag en de collectie waarin gezocht wordt dan de andere methodes die we onderzocht hebben, dus we zijn voorzichtig met het aanbevelen van deze methode totdat deze onder een groter aantal omstandigheden getest is. Een op het oog robuustere aanpak is het opdelen van de tekst of spraak in delen van een constante (vastgestelde) lengte. Op onze collectie gaf deze methode vergelijkbare resultaten met meer geavanceerde technieken zoals TextTiling of een WordNet-gebaseerde aanpak. Alle methodes die we getest hebben leidden tot een tamelijk grote reductie in MAP van relatief gezien ongeveer 10-15%. Het verbeteren van het automatisch opdelen van spraaktranscripten voor gebruik in SDR is daarom potentieel een zeer interessante manier om de kwaliteit van SDR te verbeteren.

In Hoofdstuk 4 hebben we verscheidene methodes voor het vergelijken van geordende resultatenlijsten voorgesteld (ASR-for-SDR uit Figuur 6.1). Hierbij is een belangrijke eigenschap van het door ons geïntroduceerde raamwerk dat er geen relevantieoordelen (qrels) nodig zijn. Onze aanname was dat als de resultaten van deze evaluaties een hoge lineaire correlatie hebben met (relatieve) MAP, ze een extrinsieke maat vormen voor de invloed van ruis in

ASR-transcripten of opdeelfouten op de prestaties van een zoeksysteem. Als de resultaten een hoge ordeningscorrelatie hebben met MAP, dan kunnen ze gebruikt worden om verschillende methodes of configuraties te ordenen op hun relatieve verwachte prestaties wanneer ze gebruikt worden als onderdeel van een SDR-systeem. De lineaire correlatie met MAP voor de verschillende ASR-transcripten en opdelingen bleek zeer significant te zijn, waarbij $\mathrm{RBO}_{0.95}$ een waarde van 0.997 behaalde voor transcriptieruis. Ordeningscorrelatie varieerde tussen perfect en zeer significant. Alhoewel sommige methodes een hogere lineaire correlatie hebben dan anderen, zijn de verschillen slechts zelden significant. We concluderen dat alle door ons geteste methodes geschikt zijn voor deze taak. Welke methode het beste gekozen kan worden hangt daarom vooral of van welke aspecten als meest belangrijk worden gezien voor het gebruik van het systeem. Bijvoorbeeld, de verwachte vasthoudendheid van de gebruiker bij het inspecteren van de resultaten, of eigenschappen van de interface.

Na het aantonen van de theoretische levensvatbaarheid van het voorgestelde raamwerk, hebben we in Hoofdstuk 5 de praktische aspecten bekeken. We onderzochten hoeveel referentietranscripten er nodig waren om de hoge correlaties met MAP te halen die we vonden in Hoofdstuk 4. We deden dit door op een groot aantal deelcollecties van verschillende lengtes uit een collectie van 400 uur te testen. Dit maakte het noodzakelijk om kunstmatige informatievragen (of queries) te gebruiken in plaats van de tot dan toe gebruikte queries die afkomstig zijn uit testcollecties. We ontwikkelden een systeem om kunstmatige queries te genereren, waarmee we een zelfde correlatie met MAP haalden als met echte queries. Als we genoegen nemen met een relatieve standaarddeviatie in de lineaire correlatie met MAP van (een relatief arbitrair gekozen) 3%, dan kan de kwaliteit van ASR-in-SDR-context geschat worden met slechts drie uur referentietranscripten. Dit is vergelijkbaar met wat nodig is voor de traditionele intrinsieke evaluatie met WER. We concluderen daarom dat extrinsieke evaluatie van ASR-in-SDR-context kwaliteit net zo eenvoudig is als intrinsieke evaluatie, zonder dat hiervoor meer materiaal nodig is. Alhoewel we aanbevelen om door mensen gemaakte queries te gebruiken, die ook het daadwerkelijke gebruik van het SDR-systeem aangeven, bleken kunstmatige queries een redelijk alternatief te zijn als echte queries niet verkregen kunnen worden.

Bibliography

[Ahmed et al., 1974] Ahmed, N., Natarajan, T., and Rao, K. R. (1974). Discrete cosine transform. *IEEE transactions on Computers*, 23(1):90–93.

[Allan et al., 1998] Allan, J., Carbonell, J., Doddington, G., Yamron, J., Yang, Y., Archibald, B., Beeferman, D., Berger, A., Brown, R., Carp, I., Hauptmann, A., Lafferty, J., Lavrenko, V., Liu, X., Lowe, S., Mulbregt, P. V., Papka, R., Pierce, T., Ponte, J., and Scudder, M. (1998). Topic detection and tracking pilot study final report. In *Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop*, pages 194–218.

[Azzopardi and de Rijke, 2006] Azzopardi, L. and de Rijke, M. (2006). Automatic construction of known-item finding test beds. In *Proceedings of the ACM SIGIR conference*, pages 603–604, New York, NY, USA. ACM.

[Azzopardi et al., 2007] Azzopardi, L., de Rijke, M., and Balog, K. (2007). Building simulated queries for known-item topics: an analysis using six european languages. In *Proceedings of the ACM SIGIR conference*, pages 455–462, New York, NY, USA. ACM.

[Barzilay and Elhadad, 1997] Barzilay, R. and Elhadad, M. (1997). Using lexical chains for text summarization. In *Proceedings of the ACL Workshop on Intelligent Scalable Text Summarization*, pages 10–17.

[Blest, 2000] Blest, D. (2000). Rank correlation: an alternative measure. *Australian and New Zealand Journal of Statistics*, 42:101–111.

[Brants et al., 2007] Brants, T., Popat, A. C., Xu, P., Och, F. J., and Dean, J. (2007). Large language models in machine translation. In *Proceedings of EMNLP-CoNLL*, pages 858–867.

[Burget and Hermansky, 2001] Burget, L. and Hermansky, H. (2001). Data driven design of filter bank for speech recognition. In *Text, Speech and Dialogue*, volume 2166 of *Lecture Notes in Computer Science*, pages 299–304. Springer Berlin / Heidelberg.

[Büttcher et al., 2010] Büttcher, S., Clarke, C. L. A., and Cormack, G. V. (2010). *Information Retrieval: Implementing and Evaluating Search Engines*. The MIT Press.

[Carbonell et al., 1999] Carbonell, J., Yang, Y., Lafferty, J., Brown, R. D., Pierce, T., and Liu, X. (1999). CMU report on TDT-2: Segmentation, detection and tracking. In *Proceedings of the DARPA Broadcast News Workshop*, pages 117–120. Morgan Kaufmann Publishers, Inc.

[Chelba and Acero, 2005] Chelba, C. and Acero, A. (2005). Position specific posterior lattices for indexing speech. In *Proceedings of the 43rd ACL Meeting*, pages 443–450, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Choi, 2000] Choi, F. Y. Y. (2000). Advances in domain independent linear text segmentation. In *Proceedings of the 1st North American chapter of the ACL conference*, pages 26–33, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

[Cieri et al., 1999] Cieri, C., Graff, D., Liberman, M., Martey, N., and Strassel, S. (1999). The TDT-2 text and speech corpus. In *Proceedings of the DARPA Broadcast News Workshop*, pages 57–60. Morgan Kaufmann Publishers Inc.

[Clarkson and Robinson, 1997] Clarkson, P. and Robinson, A. J. (1997). Language model adaptation using mixtures and an exponentially decaying cache. In *Proceedings of ICASSP*, pages 799–802.

[Cleverdon and Keen, 1966] Cleverdon, C. and Keen, M. (1966). Factors determining the performance of indexing systems, volume 2. *The College of Aeronautics*.

[Davis and Mermelstein, 1980] Davis, S. and Mermelstein, P. (1980). Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 28(4):357–366.

[Despres et al., 2009] Despres, J., Fousek, P., Gauvain, J.-L., Gay, S., Josse, Y., Lamel, L., and Messaoudi, A. (2009). Modeling northern and southern varieties of Dutch for STT. In *Proceedings of Interspeech*, pages 96–99.

[Dharanipragada et al., 1999] Dharanipragada, S., Franz, M., Mccarley, J. S., Roukos, S., and Ward, T. (1999). Story segmentation and topic detection in the broadcast news domain. In *Proceedings of the DARPA Broadcast News Workshop*.

[Eide and Gish, 1996] Eide, E. and Gish, H. (1996). A parametric approach to vocal tract length normalization. In *Proceedings of ICASSP*, pages 346–348, Washington, DC, USA. IEEE Computer Society.

[Evermann and Woodland, 2000] Evermann, G. and Woodland, P. (2000). Large vocabulary decoding and confidence estimation using word posterior probabilities. In *Proceedings of ICASSP*, pages 2366–2369.

[Fagin et al., 2003] Fagin, R., Kumar, R., and Sivakumar, D. (2003). Comparing top k lists. In *Proceedings of the 14th ACM SIAM symposium on Discrete algorithms*, pages 28–36, Philadelphia, PA, USA. Society for Industrial and Applied Mathematics.

[Fellbaum, 1998] Fellbaum, C., editor (1998). *WordNet: An Electronic Lexical Database (Language, Speech, and Communication)*. The MIT Press.

[Fiscus and Doddington, 2002] Fiscus, J. G. and Doddington, G. R. (2002). *Topic detection and tracking evaluation overview*, pages 17–31. Kluwer Academic Publishers, Norwell, MA, USA.

[Franz et al., 1999] Franz, M., Mccarley, J. S., Ward, T., and j. Zhu, W. (1999). Segmentation and detection at IBM: Hybrid statistical models and two-tiered clustering. In *Proceedings of the TDT-3 Workshop*.

[Garofolo et al., 2000a] Garofolo, J., Lard, J., and Voorhees, E. (2000a). Spoken document retrieval track slides.

[Garofolo et al., 2000b] Garofolo, J. S., Auzanne, C. G. P., and Voorhees, E. M. (2000b). The trec spoken document retrieval task: A success story. In *Proceedings of RIAO: Content Based Multimedia Information Access Conference*, Paris, France.

[Garofolo et al., 1998] Garofolo, J. S., Voorhees, E. M., Auzanne, C. G. P., Stanford, V. M., and Lund, B. A. (1998). TREC-7 spoken document retrieval track overview and results. In *Proceedings of TREC*.

[Garofolo et al., 1997] Garofolo, J. S., Voorhees, E. M., Stanford, V. M., and Spärck-Jones, K. (1997). TREC-6 spoken document retrieval track overview and results. In *Proceedings of TREC*, pages 83–91.

[Gauvain et al., 2000] Gauvain, J.-L., Lamel, L., Barras, C., Adda, G., and de Kercadio, Y. (2000). The LIMSI SDR system for TREC-9. In *Proceedings of TREC*.

[Gauvain and Lee, 1994] Gauvain, J.-L. and Lee, C.-H. (1994). Maximum a posteriori estimation for multivariate gaussian mixture observations of Markov chains. *IEEE Transactions on Speech and Audio Processing*, 2:291–298.

[Gottlieb et al., 1998] Gottlieb, A. H., Gottlieb, H., Bowers, B., Bowers, B., and Gottlieb, A. (1998). *1,000 Years, 1,000 People: Ranking the Men and Women Who Shaped the Millennium*. Kodansha America.

[Graff et al., 1999] Graff, D., Cieri, C., Strassel, S., and Martey, N. (1999). The tdt-3 text and speech corpus. In *Proceedings of DARPA Broadcast News Workshop*, pages 57–60. Morgan Kaufmann Publishers Inc.

[Greiff et al.,] Greiff, W., Hurwitz, L., and Merlino, A. The Mitre TDT-3 segmentation system. In *Proceedings of the TDT-3 Conference*.

[Harman and Voorhees, 2005] Harman, D. K. and Voorhees, E. M., editors (2005). *TREC: experiment and evaluation in information retrieval*. MIT Press, Cambridge, Mass.

[Hearst, 1994] Hearst, M. (1994). Multi-paragraph segmentation of expository text. In *Proceedings of 32nd. ACL Meeting*, pages 9–16, New Mexico State University, Las Cruces, New Mexico.

[Hearst, 1997] Hearst, M. A. (1997). Texttiling: segmenting text into multi-paragraph subtopic passages. *Computational Linguistics*, 23(1):33–64.

[Hermansky, 1990] Hermansky, H. (1990). Perceptual linear predictive (PLP) analysis of speech. *Journal of the Acoustical Society of America*, 87(4):1738–1752.

[Hermansky and Morgan, 1994] Hermansky, H. and Morgan, N. (1994). RASTA processing of speech. *IEEE transactions on Speech and Audio Processing*, 2(4):578–589.

[Hiemstra, 2001] Hiemstra, D. (2001). *Using Language Models for Information Retrieval*. PhD thesis, University of Twente, Enschede.

[Jiang and Conrath, 1997] Jiang, J. J. and Conrath, D. W. (1997). Semantic similarity based on corpus statistics and lexical taxonomy. In *Proceedings of ROCLING X*, pages 19–33.

[Johnson et al., 1999] Johnson, S., Jourlin, P., Moore, G., Spärck-Jones, K., and Woodland, P. (1999). The Cambridge University Spoken Document Retrieval system. In *Proceedings of ICASSP*, pages 49–52.

[Johnson et al., 2000] Johnson, S. E., Jourlin, P., Spärck-Jones, K., and Woodland, P. C. (2000). Spoken Document Retrieval for TREC-9 at Cambridge University. In *Proceedings of TREC*.

[Kendall, 1938] Kendall, M. G. (1938). A new measure of rank correlation. *Biometrika*, 30(1/2):81–93.

[Kuhn et al., 1998] Kuhn, R., Nguyen, P., Junqua, J.-C., Goldwasser, L., Niedzielski, N., Fincke, S., Field, K., and Contolini, M. (1998). Eigenvoices for speaker adaptation. In *Proceedings of ICSLP*, pages 1771–1774.

[Lee et al., 1990] Lee, K., Hon, H., and Reddy, R. (1990). An overview of the sphinx speech recognition system. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 38(1):35–45.

[Leggetter and Woodland, 1995] Leggetter, C. and Woodland, P. (1995). Maximum likelihood linear regression for speaker adaptation of continuous density hidden markov models. *Computer Speech and Language*, 9(2):171–185.

[Levenshtein, 1966] Levenshtein, V. I. (1966). Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*, 10(8):707–710.

[Macherey et al., 2003] Macherey, W., Viechtbauer, J., and Ney, H. (2003). Probabilistic aspects in Spoken Document Retrieval. *EURASIP Journal on Applied Signal Processing*, 2003(2):115–127.

[Mangu et al., 2000] Mangu, L., Brill, E., and Stolcke, A. (2000). Finding consensus among words: lattice-based word error minimisation. *Computer Speech and Language*, pages 373–400.

[Manning, 1998] Manning, C. D. (1998). Rethinking text segmentation models: An information extraction case study. Technical report, University of Sydney.

[Manning et al., 2008] Manning, C. D., Raghavan, P., and Schtze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA.

[Manning and Schutze, 1999] Manning, C. D. and Schutze, H. (1999). *Foundations of Statistical Natural Language Processing*. The MIT Press.

[Marlow et al., 2006] Marlow, C., Naaman, M., Boyd, D., and Davis, M. (2006). Ht06, tagging paper, taxonomy, flickr, academic article, to read. In *Proceedings of the Seventeenth Conference on Hypertext and Hypermedia*, pages 31–40. ACM Press.

[Mittendorf and Schuble, 1994] Mittendorf, E. and Schuble, P. (1994). Document and passage retrieval based on hidden markov models. In *Proceedings of the ACM SIGIR Conference*, pages 318–327.

[Oard et al., 2006] Oard, D., Demner-Fushman, D., Hajič, J., Ramabhadran, B., Gustman, S., Byrne, W., Soergel, D., Dorr, B., Resnik, P., and Picheny, M. (2006). Cross-language access to recorded speech in the MALACH project. In Sojka, P., Kopecek, I., and Pala, K., editors, *Text, Speech and Dialogue*, volume 2448 of *Lecture Notes in Computer Science*, pages 197–212. Springer Berlin / Heidelberg.

[Page et al., 1999] Page, L., Brin, S., Motwani, R., and Winograd, T. (1999). The PageRank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab.

[Peters and Becker, 2009] Peters, I. and Becker, P. (2009). *Folksonomies: Indexing and Retrieval in Web 2.0*. Knowledge & information : studies in information science. De Gruyter/Saur.

[Porter, 1980] Porter, M. F. (1980). An algorithm for suffix stripping. *Program 14*, pages 130–137.

[Rabiner and Juang, 2003] Rabiner, L. and Juang, B. (2003). An introduction to hidden Markov models. *ASSP Magazine, IEEE*, 3(1):4–16.

[Rabiner and Juang, 1993] Rabiner, L. and Juang, B.-H. (1993). *Fundamentals of speech recognition*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

[Renals and Hochberg, 1996] Renals, S. and Hochberg, M. (1996). Efficient evaluation of the LVCSR search space using the Noway decoder. In *Proceedings of ICASSP*, pages 149–152. IEEE.

[Robertson and Walker, 1994] Robertson, S. E. and Walker, S. (1994). Some simple effective approximations to the 2-Poisson model for probabilistic weighted retrieval. In *Proceedings of the ACM SIGIR Conference*, pages 232–241. Springer-Verlag.

[Rosenberg et al., 1994] Rosenberg, A. E., Lee, C.-H., and Soong, F. K. (1994). Cepstral channel normalization techniques for HMM-based speaker verification. In *Proceedings of ICASSP*, pages 1835–1838.

[Salton, 1971] Salton, G. (1971). *The SMART Retrieval System - Experiments in Automatic Document Processing*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

[Salton et al., 1993] Salton, G., Allan, J., and Buckley, C. (1993). Approaches to passage retrieval in full text information systems. In *Proceedings of the ACM SIGIR conference*, pages 49–58, New York, NY, USA. ACM.

[Salton et al., 1975] Salton, G., Wong, A., and Yang, C. S. (1975). A vector space model for automatic indexing. *Communications of the ACM*, 18:613–620.

[Saraclar, 2004] Saraclar, M. (2004). Lattice-based search for spoken utterance retrieval. In *In Proceedings of HLT-NAACL*, pages 129–136.

[Singhal and Pereira, 1999] Singhal, A. and Pereira, F. (1999). Document expansion for speech retrieval. In *Proceedings of the ACM SIGIR Conference*, pages 34–41. ACM Press.

[Spärck-Jones et al., 2000] Spärck-Jones, K., Walker, S., and Robertson, S. E. (2000). A probabilistic model of information retrieval: development and comparative experiments. parts 1 and 2. *Information Processing and Management*, 36(6):779–840.

[Stokes et al., 2004] Stokes, N., Carthy, J., and Smeaton, A. F. (2004). SeLeCT: a lexical cohesion based news story segmentation system. *AI Communications - STAIRS 2002*, 7(1):3–12.

[Stolcke, 2002] Stolcke, A. (2002). SRILM - an extensible language modeling toolkit. In *Proceedings of ICSLP*, pages 901–904.

[van der Werff, 2010] van der Werff, L. (2010). Story segmentation for speech transcripts in sparse data conditions. In *Proceedings of the ACM Multimedia SSCS Workshop*, ACM Multimedia, pages 33–38, New York. ACM. ISBN=978-1-4503-0162-6.

[van der Werff et al., 2007] van der Werff, L., Heeren, W., Ordelman, R., and de Jong, F. (2007). Radio oranje: Enhanced access to a historical spoken word collection. In *Proceedings of the CLIN meeting*, number 7, pages 207–218, Utrecht. Landelijke Onderzoekschool Taalwetenschap.

[van der Werff et al., 2011] van der Werff, L., Kraaij, W., and de Jong, F. (2011). Speech transcript evaluation for information retrieval. In *Proceedings of Interspeech*, pages 1525–1528. International Speech Communication Association.

[van der Werff and Heeren, 2007] van der Werff, L. B. and Heeren, W. F. L. (2007). Evaluating ASR output for information retrieval. In *Proceedings of the ACM SIGIR SSCS Workshop*, pages 7–14.

[Viterbi, 1967] Viterbi, A. (1967). Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, 13(2):260–269.

[Voorhees, 1999] Voorhees, E. M. (1999). The TREC-8 Question Answering track report. In *Proceedings of TREC-8*, pages 77–82.

[Voorhees, 2000] Voorhees, E. M. (2000). Variations in relevance judgments and the measurement of retrieval effectiveness. *Information Processing and Management*, 36(5):697–716.

[Voorhees, 2002] Voorhees, E. M. (2002). The philosophy of information retrieval evaluation. In *Revised Papers from the Second Workshop of the Cross-Language Evaluation Forum on Evaluation of Cross-Language Information Retrieval Systems*, CLEF '01, pages 355–370, London, UK. Springer-Verlag.

[Voorhees and Harman, 1998] Voorhees, E. M. and Harman, D. (1998). Overview of the seventh Text REtrieval Conference (TREC-7). In *Proceedings of TREC-7*, pages 1–24.

[Voorhees and Harman, 2000a] Voorhees, E. M. and Harman, D. (2000a). Overview of the ninth Text REtrieval Conference (TREC-9). In *In Proceedings of TREC-9*, pages 1–14.

[Voorhees and Harman, 2000b] Voorhees, E. M. and Harman, D. (2000b). Overview of the sixth Text REtrieval Conference (TREC-6). *Information Processing and Management*, 36(1):3–35.

[Voorhees and Harman, 2005] Voorhees, E. M. and Harman, D. K. (2005). *TREC: Experiment and Evaluation in Information Retrieval*. Digital Libraries and Electronic Publishing. MIT Press.

[Voorhees and Tice, 2000] Voorhees, E. M. and Tice, D. M. (2000). Building a Question Answering test collection. In *Proceedings of the ACM SIGIR Conference*, pages 200–207.

[Wayne, 2000] Wayne, C. L. (2000). Multilingual Topic Detection and Tracking: Successful research enabled by corpora and evaluation. In *Proceedings of LREC*.

[Webber et al., 2010] Webber, W., Moffat, A., and Zobel, J. (2010). A similarity measure for indefinite rankings. *ACM Transactions on Information Systems*, 28:20:1–20:38.

[Wu and Crestani, 2003] Wu, S. and Crestani, F. (2003). Methods for ranking information retrieval systems without relevance judgments. In *Proceedings of the 2003 ACM symposium on Applied computing*, SAC '03, pages 811–816, New York, NY, USA. ACM.

[Yilmaz et al., 2008] Yilmaz, E., Aslam, J. A., and Robertson, S. (2008). A new rank correlation coefficient for information retrieval. In *Proceedings of the ACM SIGIR conference*, pages 587–594, New York, NY, USA. ACM.

[Young and Chase, 1998] Young, S. J. and Chase, L. L. (1998). Speech recognition evaluation: a review of the U.S. CSR and LVCSR programmes,. *Computer Speech & Language*, 12(4):263–279.

[Young et al., 2006] Young, S. J., Kershaw, D., Odell, J., Ollason, D., Valtchev, V., and Woodland, P. (2006). *The HTK Book Version 3.4*. Cambridge University Press.