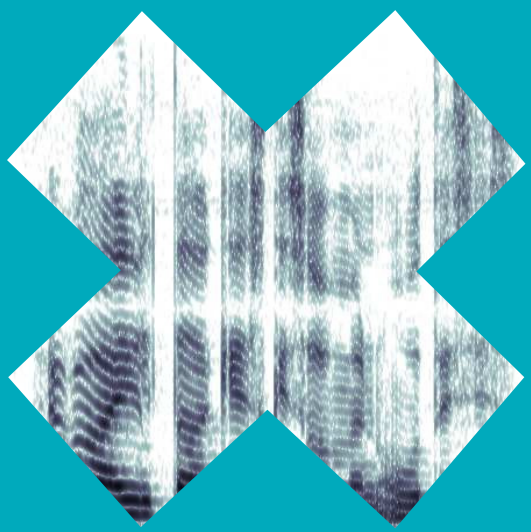
Department of Information and Computer Science

Morph-Based Speech Retrieval: Indexing Methods and Evaluations of Unsupervised Morphological Analysis

Ville T. Turunen



A!

DOCTORAL DISSERTATIONS

Aalto University publication series
DOCTORAL DISSERTATIONS 97/2012

Morph-Based Speech Retrieval: Indexing Methods and Evaluations of Unsupervised Morphological Analysis

Ville T. Turunen

Doctoral dissertation for the degree of Doctor of Science in Technology to be presented with due permission of the School of Science for public examination and debate in Auditorium AS1 at the Aalto University School of Science (Espoo, Finland) on the 24th of August 2012 at 12 noon.

Aalto University
School of Science
Department of Information and Computer Science

Supervising professor
Prof. Erkki Oja

Thesis advisor
Dr. Mikko Kurimo

Preliminary examiners
Dr. Gareth J. F. Jones, Dublin City University, Ireland
Prof. Kalervo Järvelin, University of Tampere, Finland

Opponent
Prof. Murat Saraçlar, Boğaziçi University, Turkey

Aalto University publication series
DOCTORAL DISSERTATIONS 97/2012

ISBN 978-952-60-4717-1 (printed)
ISBN 978-952-60-4718-8 (pdf)
ISSN-L 1799-4934
ISSN 1799-4934 (printed)
ISSN 1799-4942 (pdf)
http://urn.fi/URN:ISBN:978-952-60-4718-8

Unigrafia Oy
Helsinki 2012

Finland

441 697
Printed matter

Aalto University

Abstract

Aalto University, P.O. Box 11000, FI-00076 Aalto www.aalto.fi

Author
Ville T. Turunen

Name of the doctoral dissertation
Morph-Based Speech Retrieval: Indexing Methods and Evaluations of Unsupervised Morphological Analysis

Publisher School of Science

Unit Department of Information and Computer Science

Series Aalto University publication series DOCTORAL DISSERTATIONS 97/2012

Field of research Language Technology

Manuscript submitted 24 January 2012 **Date of the defence** 24 August 2012

Permission to publish granted (date) 25 June 2012 **Language** English

☐ **Monograph** ☒ **Article dissertation (summary + original articles)**

Abstract

Speech retrieval enables users to find information in collections of spoken material. Automatic speech recognition (ASR) is used to transform the spoken words into text, and information retrieval (IR) methods are used for searching. Traditional ASR systems have a predefined vocabulary of words, and any word that is out-of-vocabulary (OOV) can not be recognized. Typically, rare words are excluded, which is problematic for retrieval, because query words are often rare words such as proper names. The limited vocabulary is especially problematic for languages such as Finnish that have a very large number of distinct word forms.

In this thesis, morpheme-like subword units are used for speech recognition and retrieval. The subword units, referred to as morphs, are discovered using a data driven method that learns morphological structure from text data. Using this approach, it is possible to recognize any word in speech, even a word that was not in the training data, as a sequence of morphs. A rule-based morphological analyzer could be used to find base forms of the recognized words for indexing. However, the vocabulary of the analyzer is also limited, and recognition errors cause further problems for the analyzer. Instead, in this work, morphs are used as index terms as well.

In Finnish speech retrieval experiments, the morph-based approach is compared to using word-based language models in ASR, and to using base forms in retrieval. Also, morphs are compared for story segmentation of speech. The results show that morph-based language models clearly outperform word-based models in retrieval performance. As index terms, using morphs is about as efficient as using base forms, but combining the two approaches is better than either alone, especially when there are a high proportion of unseen words in the queries. The effect of unoptimal morph segmentations is reduced by using alternative morph segmentations of query words and by using latent semantic indexing.

Even if the morph deemed most likely by the ASR is incorrect, it is possible that the correct one is among the candidates the ASR considers. Utilizing the candidates in retrieval can improve performance. In this thesis, a representation of ASR hypotheses called confusion network is used for extracting alternative recognition results. A rank-based weighting of index terms is proposed, and found to outperform posterior probability based weighting.

This thesis also studies evaluation metrics for unsupervised morphological analysis methods. Application evaluations such as speech retrieval are time consuming and cannot be used during method development. Different linguistic evaluation metrics have been proposed and are compared in this thesis by e.g. correlating the metrics to the results of application performance.

Keywords Speech retrieval, spoken document retrieval, subword indexing, morphemes, out-of-vocabulary, confusion networks, morphological analysis

ISBN (printed) 978-952-60-4717-1 **ISBN (pdf)** 978-952-60-4718-8

ISSN-L 1799-4934 **ISSN (printed)** 1799-4934 **ISSN (pdf)** 1799-4942

Location of publisher Espoo **Location of printing** Helsinki **Year** 2012

Pages 228 **urn** http://urn.fi/URN:ISBN:978-952-60-4718-8

Aalto-yliopisto

Tiivistelmä

Aalto-yliopisto, PL 11000, 00076 Aalto www.aalto.fi

Tekijä
Ville T. Turunen

Väitöskirjan nimi
Morfeihin perustuva puhetiedonhaku: indeksointimenetelmiä sekä ohjaamattoman morfologisen analyysin evaluaatioita

Julkaisija Perustieteiden korkeakoulu

Yksikkö Tietojenkäsittelytieteen laitos

Sarja Aalto University publication series DOCTORAL DISSERTATIONS 97/2012

Tutkimusala Kieliteknologia

Käsikirjoituksen pvm 24.01.2012 **Väitöspäivä** 24.08.2012

Julkaisuluvan myöntämispäivä 25.06.2012 **Kieli** Englanti

☐ **Monografia** ☒ **Yhdistelmäväitöskirja (yhteenveto-osa + erillisartikkelit)**

Tiivistelmä

Puhetiedonhaku mahdollistaa tiedon löytämisen puhuttua aineistoa sisältävistä kokoelmista. Puheentunnistusta käytetään muuttamaan puhutut sanat tekstiksi, ja tiedonhakumenetelmiä käytetään tunnistustekstistä etsimiseen. Perinteiset tunnistusjärjestelmät sisältävät ennalta määrätyn sanaston, jolloin sanaston ulkopuoliset sanat jäävät aina tunnistumatta oikein. Yleensä harvinaiset sanat jätetään pois, mikä on ongelmallista tiedonhaun kannalta, koska hakusanat ovat usein harvinaisia sanoja, kuten erisnimiä. Rajoitettu sanasto on erityisen ongelmallista kielille, joissa on runsaasti sanamuotoja, kuten suomelle.

Tässä väitöskirjassa käytetään morfeemien kaltaisia sananosia tunnistukseen ja tiedonhakuun. Nämä morfeiksi kutsutut osat löydetään käyttäen ohjaamatonta menetelmää, joka oppii morfologista rakennetta tekstistä. Yhdistelemällä morfeja on mahdollista tunnistaa puheesta mikä tahansa sana, jopa sana, jota ei tavattu opetusaineistossa. Indeksoinnissa voidaan käyttää perusmuotoja, jotka saadaan sääntöpohjaisella morfologisella analyysaattorilla. Tällaisen analysaattorin käyttämä sanavarasto on kuitenkin rajoitettu, ja lisäksi tunnistusvirheet haittaavat sen toimintaa. Perusmuotojen sijaan tässä työssä käytetään morfeja myös indeksoinnissa.

Suomenkielisissä puhetiedonhakutesteissä verrataan morfimenetelmää perinteisiin sanakielimalleihin tunnistuksessa ja perusmuotoihin tiedonhaussa. Tiedonhakutulosten perusteella morfikielimallit ovat selvästi parempia kuin sanakielimallit. Indeksoinnissa morfien käyttö on likimäärin yhtä tehokasta kuin perusmuotojen käyttö, mutta menetelmien yhdistäminen on tehokkainta, erityisesti silloin, kun opetustekstin ulkopuolisten sanojen osuus hakusanoista on suuri. Lisäksi epäoptimaalisten morfisegmenttien vaikutus vähenee, kun käytetään vaihtoehtoisia morfisegmentaatioita tai latenttia semanttista indeksointia.

Vaikka morfi, joka tunnistimen mielestä on todennäköisin, on virheellinen, voi oikea morfi olla tunnistimen harkitsemien vaihtoehtojen joukossa. Näitä vaihtoehtoja voi hyödyntää haussa. Tässä työssä tunnistusvaihtoehdot esitetään konfuusioverkko-nimisessä rakenteessa. Vaihtoehtojen painottaminen niiden käänteisen paremmuusjärjestyksen mukaan havaitaan paremmaksi kuin painottaminen todennäköisyyden mukaan.

Tässä väitöskirjassa tutkitaan myös evaluointimenetelmiä, joilla voi mitata ohjaamattomien morfologisten analyysimenetelmien toimintaa. Sovellusevaluaatiot, kuten puhetiedonhaku, ovat aikaavieviä eikä niitä voi käyttää kehitysvaiheen aikana. Erilaisia lingvistisiä evaluaatiomenetelmiä on ehdotettu, ja niitä verrataan esimerkiksi korreloimalla niiden tuloksia suorituskykyyn sovelluksissa.

Avainsanat Puhetiedonhaku, sananosat, morfeemi, konfuusioverkko, morfologinen analyysi

ISBN (painettu) 978-952-60-4717-1 **ISBN (pdf)** 978-952-60-4718-8

ISSN-L 1799-4934 **ISSN (painettu)** 1799-4934 **ISSN (pdf)** 1799-4942

Julkaisupaikka Espoo **Painopaikka** Helsinki **Vuosi** 2012

Sivumäärä 228 **urn** http://urn.fi/URN:ISBN:978-952-60-4718-8

Preface

This work has been carried out in the Adaptive Informatics Research Centre (AIRC) of Department of Information and Computer Science, Aalto University School of Science. The work has been funded by the Graduate School in Computational Methods of Information Technology (ComMIT), AIRC, and the Tivit Next Media program (financed by TEKES – the Finnish Funding Agency for Technology and Innovation). I am also grateful for the personal grants from the Finnish Foundation for Economic and Technology Sciences – KAUTE, the Finnish Foundation for Technology Promotion (TES) and the Emil Aaltonen Foundation.

This thesis would never have come about without the help and encouragement from several people. I wish to thank the instructor of my thesis, Dr. Mikko Kurimo, for the opportunity to work in the speech group, for his guidance and support in conducting research, and for co-authoring most of the publications in this thesis. I am also grateful to my supervisor, Prof. Erkki Oja, for his support for my work, and, as the head of AIRC, for creating a working environment where it has been a pleasure and a privilege to do research.

I also want to thank the official pre-examiners of my thesis, Dr. Gareth J. F. Jones and Prof. Kalervo Järvelin. Their insightful comments helped me to improve the quality of this work substantially.

I wish to express gratitude to the co-authors of the publications of this thesis. Especially, I thank Sami Virpioja, who was the major contributor in the part of work that deals with evaluation metrics for morphological analysis. I also thank the rest of the former and current members of the speech group. The efforts by the group for creating the recognition system have been tremendous. Janne Pylkkönen and Dr. Teemu Hirsimäki deserve a special mention for patiently explaining difficult concepts in speech recognition, helping me with practical issues with the recognizer,

as well as giving ideas for research.

I also wish to thank people who have contributed in creating the resources used in this thesis. I thank Inger Ekman and the Department of Information Studies and Interactive Media at the University of Tampere for providing one of the speech retrieval corpora used in this thesis. For the other corpus, I thank the Finnish Broadcasting Company (YLE) for providing the material, and Niklas Raatikainen for assisting with creating the corpus. I am also grateful to the Cross-Language Evaluation Forum that provided the text corpora used in this thesis.

Most of all, I wish to thank Taina.

Helsinki, July 13, 2012,

Ville T. Turunen

Contents

List of Publications

This thesis consists of an overview and of the following publications which are referred to in the text by their Roman numerals.

I Mikko Kurimo, Ville Turunen and Inger Ekman. An evaluation of a spoken document retrieval baseline system in Finnish. In *Proceedings of the 8th International Conference on Spoken Language Processing (Interspeech 2004 – ICSLP)*, Jeju Island, Korea, pp. 1585–1588, October 2004.

II Mikko Kurimo and Ville Turunen. To recover from speech recognition errors in spoken document retrieval. In *Proceedings of the 9th European Conference on Speech Communication and Technology (Interspeech 2005 – Eurospeech)*, Lisbon, Portugal, pp. 605–608, September 2005.

III Ville T. Turunen and Mikko Kurimo. Using latent semantic indexing for morph-based spoken document retrieval. In *Proceedings of the 9th International Conference on Spoken Language Processing (Interspeech 2006 – ICSLP)*, Pittsburgh PA, USA, pp. 341–344, September 2006.

IV Ville T. Turunen and Mikko Kurimo. Indexing confusion networks for morph-based spoken document retrieval. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Amsterdam, The Netherlands, pp. 631–638, July 2007.

V Ville T. Turunen. Reducing the effect of OOV query words by using

morph-based spoken document retrieval. In *Proceedings of the 9th Annual Conference of the International Speech Communication Association (Interspeech 2008)*, Brisbane, Australia, pp. 2158–2161, September 2008.

VI Ville T. Turunen and Mikko Kurimo. Speech retrieval from unsegmented Finnish audio using statistical morpheme-like units for segmentation, recognition, and retrieval. *ACM Transactions on Speech and Language Processing*, Vol. 8, No. 1, pp. 1–25, October 2011.

VII Sami Virpioja, Ville T. Turunen, Sebastian Spiegler, Oskar Kohonen and Mikko Kurimo. Empirical comparison of evaluation methods for unsupervised learning of morphology. *Traitement Automatique des Langues*, Vol. 52, No. 2, pp. 45–90, 2011.

Author's Contribution

Publication I: "An evaluation of a spoken document retrieval baseline system in Finnish"

The author was responsible for implementing the experimental part of the work: running the speech recognition and retrieval experiments. The author also took part in analyzing the results and writing the article.

Publication II: "To recover from speech recognition errors in spoken document retrieval"

The author was responsible for implementing the experimental part of the work: running the speech recognition and retrieval experiments. The author also took part in analyzing the results and writing the article.

Publication III: "Using latent semantic indexing for morph-based spoken document retrieval"

The author was responsible for designing and implementing the experimental setup, and analyzing the results. The author was also the major contributor of the writing.

Publication IV: "Indexing confusion networks for morph-based spoken document retrieval"

The author was responsible for designing and implementing the experimental setup, and analyzing the results. The author was also the major contributor of the writing.

Publication V: "Reducing the effect of OOV query words by using morph-based spoken document retrieval"

The author was solely responsible for the paper: designing and implementing the experimental setup, analyzing the results, and writing the article.

Publication VI: "Speech retrieval from unsegmented Finnish audio using statistical morpheme-like units for segmentation, recognition, and retrieval"

The author was responsible for designing and implementing the experimental setup, and analyzing the results. The author was also the major contributor of the writing.

Publication VII: "Empirical comparison of evaluation methods for unsupervised learning of morphology"

The author provided the information retrieval results used in the Publication, wrote Sections 3.2.1, and 4.2, and took part in analyzing the results on metric correlations.

List of Abbreviations

AM	Acoustic model.
AP	Average precision.
ASR	Automatic speech recognition.
CL	Confidence level (weighting).
CMS	Cepstral mean subtraction.
CN	Confusion network.
CoMMA	Co-occurrence-based metric for morphological analysis.
DCT	Discrete cosine transform.
EMMA	Evaluation metric for morphological analysis.
ETF	Expected term frequency.
FFT	Fast Fourier transform.
FST	Finite state transducer.
GAP	Generalized average precision.
GMM	Gaussian mixture model.
HMM	Hidden Markov model.
IDF	Inverse document frequency.
IR	Information retrieval.
IV	In-vocabulary.
LER	Letter error rate.
LM	Language model.
LSI	Latent semantic indexing.
LVCSR	Large vocabulary continuous speech recognition.
LVQ	Learning vector quantization.
MAP	Maximum a posteriori.
MAP	Mean average precision.

MDL	Minimum description length.
MFCC	Mel-frequency cepstral coefficients.
MGAP	Mean generalized average precision.
ML	Maximum likelihood.
MLLT	Maximum likelihood linear transform.
MPFE	Minimum phone frame error.
NLP	Natural language processing.
OOV	Out-of-vocabulary (word).
PER	Phone error rate.
PLSA	Probabilistic latent semantic analysis.
PSPL	Position specific posterior lattice.
Q-OOV	Query out-of-vocabulary (rate).
RT	Real-time.
RUR	Ranked utterance retrieval.
SDR	Spoken document retrieval.
SMT	Statistical machine translation.
SOM	Self-organizing map.
STD	Spoken term detection.
SUR	Spoken utterance retrieval.
SVD	Singular value decomposition.
TER	Term error rate.
TF	Term frequency.
TFIDF	Term frequency times inverse document frequency.
TWOL	Two-level morphology (analyzer).
WER	Word error rate.
WFST	Weighted finite state transducer.

List of Symbols

avgdl	Average document length.
$DL(D)$	The length of the document D.
c	A position in a CN.
$DF(t)$	Document frequency for term t.
D_i	Document vector for the i:th document in the collection.
d_{ij}	Term weight for the j:th term in the i:th document.
$IDF(t)$	Inverse document frequency for term t.
Λ	Parameters of an HMM.
m	Number of terms in the collection.
N	Number of documents in the collection.
O	Sequence of observations.
p	Precision.
p_k	Precision at rank k.
Q	Query vector.
q_j	Term weight for the j:th term in the query.
r	Recall.
r_k	Recall at rank k.
$OW(t)$	Offer weight for term t.
$RW(t)$	Relevance weight for term t.
S	Total number of relevant document for the query Q.
$s(t)$	Number of relevant document for the query Q that contain the term t.
t	Term.
$TF(t, D_i)$	Term frequency for the term t in the document D_i.

W	Sequence of words.
$\mathbf{X}$	Term-document matrix.

1. Introduction

Huge amounts of information in the form of audio and video recordings are produced, distributed, and stored every day. The video-sharing website YouTube alone reports that more than 60 hours of video are uploaded to the site every minute (YouTube, 2012). The Finnish National Audiovisual Archive (KAVA) stores continuously 16 radio and 12 television channels and samples almost 100 channels, resulting in 104,000 hours of radio and 92,000 hours of television recordings in the year 2010 (Mourujärvi, 2011). Clearly, there is a need for digital libraries that provide easy access to this information. While searching for textual content is commonplace for example in the form of Internet search engines, methods for searching for multimedia are still lacking. Textual data can be easily represented electronically. Typically all the words in the text are used to enable *full-text* searches on collections such as a news paper archives. For multimedia material, the electronic representation varies and digital collections have relied heavily on human generated textual descriptions, *metadata*. However, generating the metadata is costly and even with metadata available, locating specific parts within the described documents remains a time consuming task.

Speech carries a significant part of the information in multimedia content. The speech content can be accessed by combining *automatic speech recognition* (ASR) techniques with *information retrieval* (IR). Automatic speech recognition is used to create a textual representation of the speech content, and then text-based information retrieval methods are applied. The biggest complication in the process is that speech recognition accuracy is rarely perfect. Typically, at least 20% of the words in the transcription are erroneous and for more difficult material such as spontaneous speech or speech with a lot of background noise, the error rate is significantly higher.

One source of errors in ASR is the limited vocabulary of the recognizer. If a word that is spoken is not in the vocabulary of the recognizer it will always be misrecognized. Typically, a number of most common words are selected for the recognizer vocabulary. Unfortunately, from information retrieval point of view, the rest, so-called *out-of-vocabulary* (OOV) words, are often the most interesting ones. One method of countering this issue is to use subword units for recognition. In theory, it is possible to use a phoneme recognizer to transcribe any segment of speech as a sequence of phonemes. Retrieval is then performed by matching the phonetic representation of the query words to the phoneme transcriptions. The problem with this approach is that the phoneme recognizer has poorer ability to model language structure and thus the resulting error rate of a phoneme recognizer is a lot higher than for a word-based recognizer.

A compromise is to use recognition units that are smaller than words but larger than phonemes. If the units are properly selected, they will provide low error rates while limiting the vocabulary as little as possible. One option is using *morphemes* – the smallest units of language that carry a meaning. Morpheme-based approach is especially well suited for *agglutinative* languages such as Finnish. An agglutinative language is a language that forms words by joining together morphemes, and has thus a high number of distinct word forms. This means that language modeling for speech recognition is a lot more challenging. A vocabulary of all the possible word forms would grow too large for efficient recognition. Using subword units such as morphemes will allow a large coverage of the language with moderate size vocabularies. Morphemes are an attractive choice for subword language modeling since, unlike for example syllables, morphemes are associated with a meaning and they encode semantics in an intuitively appealing manner.

A speech recognizer will output the sequence of words that it considers most likely given the speech input. When the recognizer makes an error, it is possible that the correct word is among the candidates of words that the recognizer considers for that location. Thus, expanding the search of query terms to these alternative candidates has the potential to increase the recall of a speech retrieval system. However, as several candidates compete for the same location and only one of these candidates can be correct, the alternative terms have to be weighted properly so that the system does not return too many irrelevant results.

Since morphemes carry a meaning, they are an appealing choice of units

for information retrieval as well, especially if content bearing morphemes can be separated from affix morphemes. Traditionally, for information retrieval of agglutinative languages, rule-based morphological analyzers are used to transform each word to its base form in order to match the words in the queries to the words in the documents irrespective of which word forms are used. An alternative is to use unsupervised morphological analyzers that can learn morphological rules from text data and can therefore be used also for languages that have no rule-based analyzers available. Recognition errors in the ASR transcriptions may also cause the rule-based morphological analysis methods to produce spurious base forms. Further, unlike rule-based analyzers, unsupervised methods are not limited by a fixed lexicon of words.

Developing unsupervised morpheme analysis tools requires methods for estimating their performance. The usefulness of the method is ultimately decided on how well it performs in the target application task such as speech recognition or information retrieval, but such evaluations are often too time consuming to be used during development. Morphological analysis methods can be also evaluated by comparing the results to linguistic reference analyses, but evaluations like this are complicated because the unsupervised methods can generate arbitrary labels for different morpheme classes and not the same as the ones used in the linguistic references. Further, it is not certain that the linguistic analyses are the most optimal for all tasks. Thus, evaluation of morphological analysis is a difficult problem in itself.

1.1 On Speech Recognition and Information Retrieval Research in Finland

This work is a continuation of a long line of research in automatic speech recognition at the Laboratory of Computer and Information Science at Helsinki University of Technology. After organizational revisions within the university, and across universities, the work is now continued at the Department of Information and Computer Science at Aalto University School of Science. In the 1970s, Prof. Teuvo Kohonen started the work by applying neural network and other pattern recognition methods on speech signals. For example, in their PhD theses, Jalanko (1980) used subspace methods, and Torkkola (1991) neural networks for phonetic speech recognition. More conventional hidden Markov model (HMM) based phonetic

recognizer was presented by Kurimo (1997). Recognition performance was improved by using self-organizing map (SOM) and learning vector quantization (LVQ) for training the HMMs.

In the theses of Siivola (2006), Creutz (2007), and Hirsimäki (2009), the research was moved into large vocabulary continuous speech recognition (LVCSR). A research recognition system was developed, and special focus was on language modeling for agglutinative languages such as Finnish. Creutz developed data driven methods for morphological analysis of words that were also proven efficient for language modeling in speech recognition. Recently, discriminative training (Pylkkönen, 2009) and noise robust methods (Remes et al., 2011) for speech recognition have been developed.

Information retrieval has also been studied in the department, and the self-organizing map has been heavily used. SOM allows organizing document into meaningful maps for exploration and searching, and it has been applied to text (Kaski et al., 1998), spoken documents (Kurimo, 2000), images (Laaksonen et al., 2000) and videos (Sjöberg et al., 2011). Other IR research include tracking eye movements (Puolamäki et al., 2005), and accessing contextual information in an augmented reality setting (Ajanki et al., 2010). IR has also been an evaluation task in the Morpho Challenge competitions organized by the department (see e.g. (Kurimo et al., 2010a)).

At other departments in Aalto University, research related to information retrieval has been performed in the Semantic Computing Research Group (SeCo), focusing especially on machine processable semantics (see e.g. Hyvönen et al., 2005). Elsewhere in Finland, DOREMI Research Group at Helsinki University have research in e.g. IR, media monitoring, and information extraction (see e.g. Lehtonen and Doucet, 2008). Most notably, IR research in Finland has been advanced by the Finnish Information Retrieval Expert Group (FIRE) at University of Tampere, in which e.g. query expansion (Järvelin et al., 2001), cross-language IR (Pirkola et al., 2001), and image retrieval (Markkula and Sormunen, 2000) have been studied.

1.2 Scope of the Thesis

The topic of this thesis is speech retrieval, and as a related theme, evaluation metrics of unsupervised learning of morphology. In speech retrieval,

methods from speech recognition, story segmentation, and indexing and retrieval are needed. The morphology of the language has an effect on these parts of the system, and in this thesis, the use of morphs as units for each of these components is studied. Of particular interest is to study how well it is possible to retrieve "unseen" query words, words that have not been observed in the training data. The experiments are performed on Finnish data, and while Finnish has properties that make the use of traditional word-based methods particularly problematic, the methods in this thesis are likely to extend to other similar languages as well. The presented methods are unsupervised, and unsupervised methods are of special interest, because they are especially suited for less resourced languages. In development of unsupervised methods, performance metrics that are readily computable are needed. Therefore, this thesis also studies correlation of metrics for unsupervised morphological analysis to application tasks. The results will eventually benefit applications such as text or speech retrieval.

A complete speech retrieval system has other parts as well. When indexing broadcast material, the input data will have segments of non-speech material, such as music. Using audio segmentation and classification methods, the audio can be labeled by content type for more efficient browsing, and for detecting which parts of the audio to further process with a speech recognizer. Speaker segmentation and clustering (*diarization*) will also help browsing the contents. These topics are outside the scope of this thesis.

1.3 Contributions of the Thesis

The main contributions of this thesis are:

- A morph-based method for speech retrieval is presented and compared to traditional word-based approaches. Morphs are used as language modeling units, as segmentation units and as retrieval units, and in each step compared to using rule-based morphological analysis. It is shown that the morph-based approach is superior, especially when retrieving unseen query words (Publications I–VI).

- A new retrieval evaluation corpus of unsegmented Finnish audio is designed and constructed (Publication VI).

- Methods for improving retrieval performance by extracting alternative recognition candidates from confusion networks are compared (Publications IV and VI).

- The use of query expansion (Publication II), latent semantic indexing (Publication III), and alternative query word segmentation (Publication VI) for reducing the effect of allomorphy on the morph-based system is studied.

- Evaluation metrics for morphological analysis are compared empirically (Publication VII).

1.4 Structure of the Thesis

This thesis consists of an introduction and a collection of publications. The introduction aims to give a coherent presentation of the topic and the methods, as well as present the most central results from the publications. Full details of the experimental setups and results are given only in the publications, and reading the individual publications is absolutely necessary for full understanding.

In Chapter 2, commonly used speech retrieval tasks are defined, and a short review of relevant information retrieval models and evaluation metrics is presented. Chapter 3 gives the necessary linguistic background for understanding this thesis: morphological properties of languages are presented, as well as methods for morphological analysis of words. Speech recognition methods are presented in Chapter 4. In Chapter 5, the morph-based speech retrieval system is described, and experimental results are given. In Chapter 6, the use of lattices and confusion networks for improving retrieval performance is explored. In Chapter 7, the minor theme of this thesis is presented: empirical comparison of metrics of morphological analysis. Finally, in Chapter 8, the introductory part is concluded and discussed.

2. Speech Retrieval Tasks, Models and Evaluations

2.1 Retrieval Tasks

Speech retrieval is the task of finding segments that match the user's need from a collection of speech data. A number of possible applications exist, such as retrieval of voice and video mails (Brown et al., 1996) and accessing information from recordings of meetings (Morgan et al., 2001). An important application that can benefit from speech retrieval technologies is accessing broadcast material. Journalists need to search large archives for possibly very specific segments that can be used in documentaries or in news items. Archives of TV and radio material are also open for public online, and users will want to find segments that match their information need or are otherwise interesting or entertaining.

In information retrieval, the user's information need is formulated in a query and described in a *query representation* while the collection to be searched is described in a *document representation*. The most apparent choice for representing the query is using natural language. However, since digital audio is represented as a sequence of samples in its raw form, there is representation mismatch that needs to be solved in order to perform searches on a speech collection. The most common option is to convert the document collection to text in natural language with the help of an automatic speech recognition system.

For evaluation of speech retrieval methods, the problem has to be simplified and for that purpose different tasks have been defined. In one of the earliest approaches, *keyword spotting* (Foote et al., 1995), a small list of keywords is defined before search. The documents in the corpus are represented in terms of the spotted keywords. If a query requires a new keyword, the entire corpus needs to be reprocessed, which means long

delays in response time. Keyword spotting is a light-weight approach to speech retrieval but due to its limitations it has been replaced by methods that do not limit the search term to predefined words.

Spoken term detection (STD) (Fiscus et al., 2007) is the task of detecting all of the occurrences of a term in a large audio corpus. Unlike in keyword spotting, the search terms are not given beforehand. Response times need to be small, which means the audio corpus needs to be indexed before searching, without the knowledge of the search terms. The search terms are words or short sequences of words spoken consecutively. Transcriptions of the audio are used to define the reference occurrences of the terms. The search term must exactly match the transcription – substrings and different inflected forms are not considered occurrences. Typical approaches to STD combine *large vocabulary continuous speech recognition* (LVCSR) with phonetic representations of OOV terms (Mamou et al., 2007).

In *spoken document retrieval* (SDR) (Garofolo et al., 2000), the task is to find excerpts from archives of recordings of speech that are relevant to a user specified text query. The system should return relevant documents irrespective of which word forms, or which words, are used to describe the contents of the documents and the queries. In comparison to STD and keyword spotting, in SDR the queries are natural language descriptions of the information need and tend to be a lot longer than the words and phrases used in STD. STD and keyword spotting consider the task of speech retrieval from a technology centric point of view, while SDR is a user centric approach to the speech retrieval problem. It is the broadest and most challenging task and it is the one used in this thesis. On the other hand, testing SDR systems requires more human resources, because the relevance of each segment to each query needs to be assessed. However, in an SDR system, the techniques developed for STD and keyword spotting can be utilized.

Spoken utterance retrieval (SUR) (Saraçlar and Sproat, 2004) is the task of finding short snippets of speech (utterances) that contain the query words or phrases from a collection of unstructured speech data such as recordings of lectures, telephone conversations and meetings. Typically, the task uses speech collections that are more diverse than the broadcast material that is usually used for SDR. The unstructured nature of the data requires identifying more specific locations of the relevant portions. *Ranked utterance retrieval* (RUR) (Olsson and Oard, 2009) is a sim-

ilar task, but the utterances are returned in ranked order from the one deemed most likely to contain the query terms.

2.2 Retrieval Corpora

To perform information retrieval experiments, a test corpus is needed. A corpus consists of three things: (i) a collection of documents, (ii) a set of queries, and (iii) a set of relevance judgements. The queries are natural language expressions of information needs. Here, the word "document" is used in broad meaning to cover also the case of speech, where a document is any topically coherent segment of speech in a stream of audio. Each document in the collection is classified either as relevant or non-relevant with respect to each query. These so-called *ground truth* judgements of relevance are done by human assessors.

In this thesis, two Finnish speech retrieval corpora are used. The first one (Tampere), constructed at University of Tampere (Ekman, 2003), has 288 news stories read by single speaker in a quiet environment and is used in Publications III, IV, and V. Each of the news stories matches exactly one of the 17 queries. The transcription of the audio was available for reference experiments and speech recognition error rate measurements.

The small size of the Tampere-corpus made the conclusions less reliable than desired, thus another corpus (Podcast) was developed for Publication VI. The corpus has 136 hours of Finnish radio programs from the Finnish Broadcasting Company (YLE) downloaded as mp3 podcasts. Based on the associated metadata, 25 topic descriptions were formulated. The audio was not segmented into topical boundaries, and thus relevance was judged in terms of replay points, that is, the points in time where the relevant portion starts were located. Total of 451 relevant replay points were located by human assessors.

In addition to the speech corpora, a text corpus was used for reference text retrieval experiments in Publications VI and VII. The corpus (courtesy of Cross-Language Evaluation Forum (CLEF) (Agirre et al., 2009)) consisted of 55,000 documents from Finnish newspapers with a total of 4.6 million words. The corpus was associated with 50 topics and 23,000 binary relevance assessments with 413 relevant documents.

2.3 Information Retrieval Models

The goal of information retrieval is to return the documents that satisfy the user's information need. Usually, the system returns a set of documents in a ranked order starting from the document that best matches the query. The *information retrieval model* determines the process of matching documents to queries.

2.3.1 Vector Space Model

As the name suggests, in a *vector space model* (Salton et al., 1975) the documents D_i and the query Q are represented as vectors:

$$D_i = (d_{i1}, d_{i2}, \ldots, d_{im}) \quad (2.1)$$

$$Q = (q_1, q_2, \ldots, q_m) \quad (2.2)$$

Each dimension of the vectors correspond to a term, and m is the number of unique terms in the collection. The values of d_{ij} and q_j are called *term weights* and they receive a non-zero value if the corresponding term is present in the document or in the query. The similarity score between a query and a document is determined by their *cosine similarity* or the cosine of the angle between their vectors (Salton, 1971):

$$score(D_i, Q) = \cos\theta = \frac{D_i \cdot Q}{\|D_i\|\|Q\|} = \frac{\sum_{j=1}^{m} d_{ij} q_j}{\sqrt{\sum_{j=1}^{m} d_{ij}^2}\sqrt{\sum_{j=1}^{m} q_{ij}^2}} \quad (2.3)$$

Cosine similarity varies between zero (completely dissimilar) to one (completely similar). In ranked retrieval, the documents are returned in order of decreasing cosine similarity.

Term weights measure how important the term is in the corpus, and are most often determined using a *term frequency - inverse document frequency* (TFIDF) model. Term frequency $TF(t, D_i)$ is the number of times the term t appears in the document D_i. The bigger the term frequency, the better the term describes the contents of the document. Document frequency $DF(t)$ is the number of documents that contain the term t. The bigger the document frequency, the less discriminative the term t is, that is, the less important it is describing the contents of any document. With a collection of N documents, the inverse document frequency is defined as (Spärck Jones, 1972):

$$IDF(t) = \log \frac{N}{DF(t)} \quad (2.4)$$

Thus, the IDF of a rare term is high, whereas the IDF of a frequent

term is low. Finally, term weight for the term t corresponding to the j:th element in the vector of the i:th document is: $d_{ij} = TF(t, D_i) \cdot IDF(t)$.

Several variants of the TFIDF model have been proposed (Salton and Buckley, 1988). If a term occurs fifty times in a document, it is probably not fifty times as significant as the term occurring only once. The effect of high term frequency can be dampened using the logarithm by replacing the raw term frequency $TF(t,D_i)$ by (Manning et al., 2008):

$$WF(t, D_i) = \begin{cases} 1 + \log(TF(t, D_i)), & \text{if } TF(t, D_i) > 0 \\ 0, & \text{otherwise} \end{cases} \tag{2.5}$$

The term weight is now calculated by: $d_{ij} = WF(t, D_i) \cdot IDF(t)$.

Document length is another factor that has an effect. Long documents have high term frequencies as well as more unique terms. In Equation 2.3, the information on the length of the document is removed by normalizing the document vectors by their length. This way, long documents are not favored simply because they have more terms. It is possible to study the effect of document length empirically, and to plot the probability of relevance as function of document length. Similar function can be plotted for the relevance predicted by cosine similarity. It turns out that longer documents actually have a higher probability of being relevant, and that cosine similarity normalization unreasonably favors short documents (Singhal et al., 1996). In *pivoted document length normalization*, the pivot point, where the predicted and actual probability of relevance meet, is located and the cosine similarity curve is rotated about the point so that long documents are given higher score than before (Singhal et al., 1996).

2.3.2 Probabilistic Model

Another commonly used model for information retrieval is the probabilistic model, where the idea is to rank the documents D by decreasing probability of relevance R given the query Q: $P(R = 1|Q, D)$ (Robertson and Spärck Jones, 1976). Document and query vectors are now assumed binary, that is, $d_t = 1$, if the t:th term is present in the document, and $d_t = 0$ otherwise. Starting from this probability, let us derive a ranking function. The following derivation is based on (Robertson and Spärck Jones, 1976; van Rijsbergen, 1979; Manning et al., 2008). Equivalent to ranking by the probability is to rank by odds ratio $\frac{P(R=1|Q,D)}{P(R=0|Q,D)}$, since this transformation

is monotonic. Using the Bayes rule, the odds ratio becomes:

$$\frac{P(D|R=1,Q)P(R=1|Q)}{P(D|R=0,Q)P(R=0|Q)}.$$

$P(R=1|Q)$ and $P(R=0|Q)$ are independent of the document vectors and can be removed without changing the ranking. Assuming terms in the document are mutually independent, the probabilities can be expressed as a product over the terms in the document:

$$\frac{P(D|R=1,Q)}{P(D|R=0,Q)} = \prod_{t=1}^{m} \frac{P(d_t|R=1,Q)}{P(d_t|R=0,Q)}$$

Let us denote $p_t = P(d_t = 1|R=1,Q)$ and $u_t = P(d_t = 1|R=0,Q)$. Then $P(d_t = 0|R=1,Q) = 1 - p_t$ and $P(d_t = 0|R=0,Q) = 1 - u_t$. By separating based on d_t, the ranking becomes:

$$\prod_{t,d_t=1} \frac{p_t}{u_t} \cdot \prod_{t,d_t=0} \frac{1-p_t}{1-u_t} = \prod_{t,d_t=q_t=1} \frac{p_t}{u_t} \cdot \prod_{t,d_t=0,q_t=1} \frac{1-p_t}{1-u_t}.$$

In the right hand side, it was assumed that if $q_t = 0$ then $p_t = u_t$, that is, terms not occurring in the query are equally likely to appear in relevant and non-relevant documents. Let us include the query terms that appear in the document to the right product, but dividing them in the left product so that the value is left unchanged:

$$\prod_{t,d_t=q_t=1} \frac{p_t(1-u_t)}{u_t(1-p_t)} \cdot \prod_{t,q_t=1} \frac{1-p_t}{1-u_t}.$$

The right product is now constant over all documents, and can be ignored in the ranking. By taking a logarithm of the product on the left, we are left with a function called *Retrieval Status Value* (RSV) (Robertson and Spärck Jones, 1976):

$$RSV_D = \log \prod_{t,d_t=q_t=1} \frac{p_t(1-u_t)}{u_t(1-p_t)} = \sum_{t,d_t=q_t=1} \log \frac{p_t(1-u_t)}{u_t(1-p_t)} = \sum_{t,d_t=q_t=1} RW(t), \tag{2.6}$$

where $RW(t)$, called *relevance weight* of the term t, is:

$$RW(t) = \log \frac{p_t(1-u_t)}{u_t(1-p_t)} = \log \frac{p_t}{1-p_t} + \log \frac{1-u_t}{u_t}. \tag{2.7}$$

The relevance weight $RW(t)$ will be positive if the term is more likely to appear in relevant documents than in non-relevant.

p_t and u_t can be estimated based on statistics of the collection. Assuming that the set of relevant documents is known, and $s(t)$ is the number of relevant documents that have the term present and S is the total number of relevant documents, the maximum likelihood estimates are $p_t = s(t)/S$

and $u_t = (DF(t) - s(t))/(N - S)$. As before, $DF(t)$ is the number of documents that contain the term and N is the total number of documents. With these values,

$$RW(t) = \log \frac{s(t)/(S - s(t))}{(DF(t) - s(t))/(N - DF(t) - (S - s(t)))}. \tag{2.8}$$

To avoid the possibility of zeroes, it is common to apply smoothing by adding $\frac{1}{2}$ to each count:

$$RW(t) = \log \frac{(s(t) + \frac{1}{2})/(S - s(t) + \frac{1}{2})}{(DF(t) - s(t) + \frac{1}{2})/(N - DF(t) - S + s(t) + \frac{1}{2})}. \tag{2.9}$$

So far, term frequencies have been ignored in the ranking. Okapi BM25 is a related method, that uses term frequencies, and that has been proven empirically especially successful (Robertson et al., 1995). The query Q is now defined as containing terms $t_1 \dots t_n$. The BM25 score between a document D and a query Q is given as:

$$score(D, Q) = \sum_{j=1}^{n} IDF(t_j) \cdot \frac{TF(t_j, D) \cdot (k_1 + 1)}{TF(t_j, D) + k_1 \left(1 - b + b\frac{DL(D)}{\text{avgdl}}\right)}, \tag{2.10}$$

where $k_1 \geq 0$ and $0 \leq b \leq 1$ are parameters, $DL(D)$ is the length of the document D, and is avgdl the average document length. The k_1 parameter scales the term frequency. At large values of k_1, the effect is the same as when using raw term frequency, and at $k_1 = 0$ the TF is ignored and treated as a binary value. The b parameter determines how much the score is scaled by document length: $b = 0$ means no scaling and $b = 1$ full scaling. The IDF is given by Equation 2.9 by assuming that there is no relevance information, that is, $s(t) = S = 0$:

$$IDF(t) = \log \frac{N - DF(t) + \frac{1}{2}}{DF(t) + \frac{1}{2}}. \tag{2.11}$$

Note that the IDF gets negative values if the term appears in more than half of the documents. This can be avoided by the use of a stop list, or by flooring the IDF to zero.

2.3.3 Language Model Based Retrieval

In the language modeling approach, a language model M_d is estimated for each document, and the documents are ranked based on the probability of the model generating the query $P(Q|M_d)$ (Ponte and Croft, 1998). An advantage to the preceding models is that the language modeling approach can take proximity information into account simply by using language

models with order greater than one. That is, the estimated probabilities for a word occurring are conditional on a number of preceding words. The vector space and probabilistic models unrealistically assume that the query terms are independent. This is of special concern for subword-based retrieval, since the shorter the subword units used for retrieval are, the stronger the dependency between neighboring units is.

2.3.4 Latent Semantic Indexing

The preceding information retrieval models match terms in the documents with those in the query. However, the fact that there are *synonyms*, multiple words that have similar meanings, means that the terms in the query may not match the terms in a relevant document. On the other hand, a query including a *polyseme*, a word that has multiple different meanings, may also return non-relevant documents. An information retrieval method developed to overcome these problems is *latent semantic indexing* (LSI) (Deerwester et al., 1990), that is based on the idea of projecting the document vectors to a lower dimensional space, the dimensions of which are hoped to correspond to the underlying, latent, meanings. The latent dimensions are seen as the true representation, which was then partially obscured by a generation process that used different words at different locations.

LSI uses singular value decomposition (SVD) to find the least squares fitting of the term-document association matrix $\mathbf{X}$ to a lower dimensional space. The columns of $\mathbf{X}$ are the document vectors D_i, thus for a collection of N documents with m unique terms, $\mathbf{X}$ is an $m \times N$ matrix. The SVD is defined as (Deerwester et al., 1990):

$$\underset{m\times N}{\mathbf{X}} = \underset{m\times r}{\mathbf{U}} \; \underset{r\times r}{\mathbf{\Sigma}} \; \underset{r\times N}{\mathbf{V}^T}, \tag{2.12}$$

where $\mathbf{U}$ and $\mathbf{V}$ are matrices of *left* and *right singular vectors* and have orthonormal columns, $\mathbf{\Sigma}$ is a diagonal matrix of *singular values* of $\mathbf{X}$, and r is the rank of $\mathbf{X}$. The matrices $\mathbf{U}$ and $\mathbf{V}$ represent the terms and the documents in the new space. The matrix $\mathbf{\Sigma}$ has the singular values in descending order, that means the first dimension of the new space is the direction of greatest variance. The dimensionality reduction is achieved by taking only the k first rows of the matrixes (Deerwester et al., 1990):

$$\underset{m\times N}{\mathbf{X}_k} = \underset{m\times k}{\mathbf{U}_k} \; \underset{k\times k}{\mathbf{\Sigma}_k} \; \underset{k\times N}{\mathbf{V}_k^T}, \tag{2.13}$$

which gives the best rank-k approximation of $\mathbf{X}$ in the least squared sense.

For retrieval, the query must be projected to the same k-dimensional space (Deerwester et al., 1990):

$$\underset{1\times k}{\hat{Q}^T} = \underset{1\times m}{Q^T}\ \underset{m\times k}{\mathbf{U}_k}\ \underset{k\times k}{\mathbf{\Sigma}_k^{-1}} \tag{2.14}$$

Ranked retrieval is achieved by calculating the cosine similarity in the reduced space between the query vector $\hat{Q}$ and the document vectors (rows of $\mathbf{V}_k$). In the latent semantic space, a query and a document can have high cosine similarity even if they do not share any terms.

LSI has been found to improve retrieval performance, but the experiments have been performed on relatively small corpora of only thousands or tens of thousands of documents, such as in the original work of Deerwester et al. (1990) that used a corpus of 1033 documents. A survey of corpus sizes in LSI experiments can be found in (Bradford, 2008). The computational complexity and memory requirements of the SVD calculation rise as the number of documents and terms increases. On larger collections, it is possible to calculate the SVD on a subset of the documents and use the estimated matrix to project the rest of the documents to the latent space. This strategy was found to produce modest improvements in IR performance over standard vector-space model on a collection of 752k documents (Dumais, 1995). Recent improvements in computational power and SVD algorithms have made possible to calculate full LSI on larger collections. Using large collections, Atreya and Elkan (2011) report no improvements over Okapi BM25 based retrieval.

2.3.5 Query Expansion

Another approach to deal with the problem of synonyms is *query expansion*, where search terms are added to the user's initial query by using a thesaurus of related terms (Voorhees, 1994) or by inferring related terms from a corpus. In the latter, a *relevance feedback* process is used: the user's query is performed, and the user marks in the initial set of results some documents as relevant or non-relevant. Based on the feedback, terms are added to the query and the search is then repeated using the expanded query. Since the feedback process is often undesired for users, the set of relevant documents is usually not known. Instead, *blind relevance feedback* is used, where the top S returned documents are simply assumed relevant. Each term t in the top S documents is ranked using an *offer weight* ($OW(t)$) (Robertson, 1990):

$$OW(t) = s(t)RW(t), \tag{2.15}$$

where $RW(t)$ is the relevance weight for the term from Equation 2.9 and $s(t)$ is the number of documents in the S documents that are (pseudo) relevant that contain the term t. The query is expanded with a number of top ranked terms. The terms can be weighted uniformly, by their offer weight, or by some other function such as $1/rank$ (Renals et al., 2000). This query expansion method was also used in Publication II of this thesis.

Query expansion is particularly useful in speech retrieval, where OOVs and recognition errors degrade the quality of the transcripts. If a query word is not recognized correctly, a relevant document may be left unretrieved. Query expansion helps by adding query terms that have a similar meaning or a statistical relation to the original query terms. However, using the recognized audio corpus itself for query expansion is dangerous, because the expansion terms may then include recognition errors. Therefore, query expansion is performed on a parallel text corpus (*parallel blind relevance feedback*), and the expanded queries are then submitted to the original speech corpus (Woodland et al., 2000). Experimental results show that speech retrieval performance can be greatly improved by query expansion (Renals et al., 2000; Jourlin et al., 1999).

Another option is to use knowledge of word confusability by trying to mimic the mistakes the speech recognizer might have made by expanding OOV query words with similarly sounding IV (in-vocabulary) phrases (Logan and Thong, 2002).

2.4 Story Segmentation

Unlike text, speech material is not typically organized into topically coherent segments. For example, a TV news show will contain multiple unrelated stories that are not specifically marked. *Story segmentation* is the task of finding the points where the topic shifts from one story to another, and is used as a preprocessing step to facilitate speech retrieval from unsegmented material. The transitions between stories can be inferred from changes in speaking style, speaker participation, word content, and, if available, visual features in the associated video.

2.4.1 Lexical Segmentation Methods

The words that are used to describe the contents of the story are an obvious clue to determine where the story shifts. ASR can be used to access

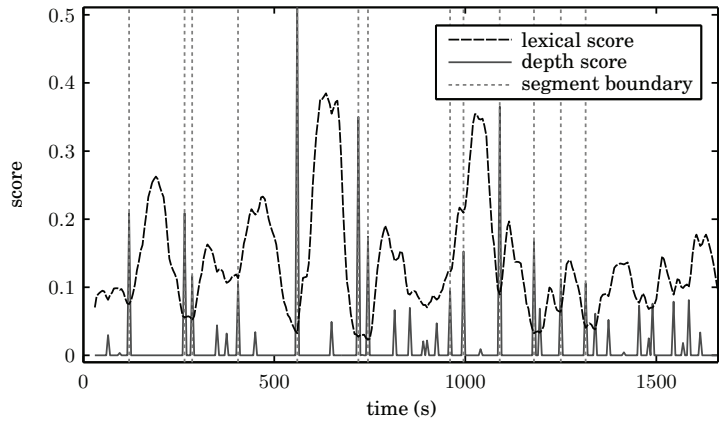


Figure 2.1. The lexical scores and the depth scores for a segment of audio. Vertical lines indicate proposed story boundaries.

the lexical content of speech material. The lexical cohesion within a story tends to be higher than the cohesion between stories. A popular method based on this idea is *TextTiling* (Hearst, 1997). It uses cosine similarity to determine the lexical similarity between two adjacent sliding windows that are moved over the text. The similarity can be plotted as a curve, and the valleys that are deep enough are assigned as story boundaries.

More precisely, the *lexical score* $ls(g)$ of the left and right windows W_l and W_r around a point g is defined as:

$$ls(g) = cos(W_l, W_r) = \frac{\sum_{i=1} w_{i,l} w_{i,r}}{\sqrt{\sum_{i=1} w_{i,l}^2 \sum_{i=1} w_{i,r}^2}}, \tag{2.16}$$

where $w_{i,l}$ and $w_{i,r}$ are term weights for the term number i in the left and right window respectively. Usually, $ls(g)$ is smoothed with a moving average filter. To determine the boundaries, each valley v in the lexical score function is compared to the left and right neighboring peaks h_l and h_r. *Depth score* ($ds(v)$) is defined as:

$$ds(v) = (ls(h_l) - ls(v)) + (ls(h_r) - ls(v)). \tag{2.17}$$

If the depth score exceeds a threshold $\theta = \mu - \alpha\sigma$ at some point, the point is assigned as a story boundary. μ and σ are the mean and standard deviation of the depth score over the document and α is a parameter. Figure 2.1 shows the lexical score, the depth score, and the proposed story boundaries for a segment of audio.

Galley et al. (2003) use *lexical chains* to determine story boundaries. A lexical chain consists of all the repetitions of a term in the text. Chains are cut to parts at locations that have a long gap between terms. The more

compact the chain is and the more frequent the term is, the stronger the chain is, that is, the more weight it is given. Cosine distance is again used to determine the lexical similarity between windows, but instead of document vectors, lexical chains are used. Similarity is high if the windows contain high number of common strong chains. Similarity is at the lowest, and story boundaries are assigned, at locations that have a high number of beginning and end points of strong chains.

Instead of calculating the similarity only between adjacent regions, the similarity between every pair of regions can be used. Coherent segments appear in the resulting *similarity matrix* as square regions along the diagonal. Reynar (1994) uses word repetition to construct the similarity matrix, called *dotplot* in this case. If the word that appears at position i appears also at position j, the element (i, j) of the dotplot receives value of 1. Choi (2000), on the other hand, uses cosine similarity between sentence pairs and instead of pure similarity values, each element of the similarity matrix receives a value that is based on the rank of the element compared to neighboring elements. In both cases, the story boundaries are determined from the similarity matrix by using a clustering method.

Certain words or phrases may indicate a change in topic. Passonneau and Litman (1997) found a correlation between a manually selected list of *cue phrases* and labeled topic boundaries. Because different domains have different set of cue phrases, learning the set from data saves the manual burden of defining the words. Beeferman et al. (1999) use an exponential model to assign probabilities of the existence of a boundary given the context. A training corpus is used to find a set of cue phrases (and other features) with a greedy search algorithm that in each step selects the most informative feature, that is, the feature that results in the biggest increase in the likelihood of the training data.

In the generative approach to story segmentation, it is viewed that there exists an underlying sequence of topics, from which the text is generated in a noisy process. Mulbregt et al. (1998) use *hidden Markov models* (HMMs) that are estimated for each topic by clustering the training data and estimating a language model for each cluster. The input text is split to fixed length sequences and the language models are used to give the *emission probabilities*, that is, for each sequence and each topic, the probability that the sequence is generated by the topic. The most likely sequence of topic states can be found using the Viterbi algorithm. Locations where there is a state transition, are the resulting story boundaries.

Term frequency based measures have the disadvantage that they do not take into account synonyms and other topically related terms. A manually built thesaurus can be used to group related words when constructing lexical chains (Morris and Hirst, 1991). Improvements in segmentation accuracy have been achieved by calculating the similarities of text regions in latent semantic space (Choi et al., 2001; Olney and Cai, 2005). Blei and Moreno (2001) use an *aspect HMM* that combines *probabilistic latent semantic analysis* (PLSA) with HMMs. In this case, the topics are latent variables, and a segment is not associated with a single topic, but a probability distribution over topics.

2.4.2 Prosodic Features for Segmentation

People often change the rhythm, stress or intonation of their speech to signal moving to a new topic. Including these *prosodic* features will aid in segmentation, especially since the lexical methods are affected by ASR errors. Shriberg et al. (2000) extracted a number of prosodic features and trained a decision tree for segmentation. It was found that long pauses and low pitch at the end of segments were the most useful features for predicting topic change in broadcast news data. However, the most indicative features depend on the nature of the data. Tür et al. (2001) combined prosodic and lexical cues using decision trees and HMMs. The best performance was achieved using a system that uses a prosodic decision tree to estimate topic change likelihoods, which were added to the HMM alongside lexical features.

In meeting data, different speakers are more active when different topics are discussed. While a speaker change in itself is not an indicator of a change in topic, the pattern of speaker activity throughout the recording can be used to infer topic changes (Galley et al., 2003; Renals and Ellis, 2003).

2.5 Evaluation metrics

2.5.1 Document Retrieval Metrics

Given a query, an information retrieval system returns a ranked list of documents. To compare the performance of different systems, the ranked list is compared against human assessed relevance information. Several

different metrics are used for evaluating system performance, the most fundamental of which are precision and recall (Manning et al., 2008). *Precision* (p) is the proportion of retrieved documents that are relevant. *Recall* (r) is the proportion of the relevant documents that are retrieved. In the case of ranked lists, these metrics are best defined in terms of some cut-off rank k. Thus, for *precision at* k (p_k) and *recall at* k (r_k) only the topmost k returned documents are considered.

By calculating precision and recall values at every rank, precision can be plotted as function of recall to give a *precision-recall curve* ($p(r)$) (Manning et al., 2008). These curves tend to be jagged, because if the document at some rank is relevant and the document at the following rank is not, precision drops while recall stays the same. Similarly, the curve jumps up and to the right, if the following document is relevant since then both precision and recall increase. A smooth curve can be achieved by using *interpolated precision-recall curve*, where precision at certain recall r is defined as the maximum precision at recall levels greater than or equal to r. For reliable estimates of precision-recall functions, the values need to be calculated and averaged over a number of queries. To reduce the number of data points, traditionally precision is calculated only at recall levels of $0.0, 0.1, 0.2, \ldots, 1.0$. Precision values at each of these points are calculated for each query and then averaged to form the so called *11-point interpolated average precision* used for example in the TREC evaluations (Voorhees and Harman, 1998).

Precision-recall curves are informative, but sometimes it is necessary to describe the performance of the system in terms of a single number. *Mean average precision* (MAP) is the most often used for this purpose. Let S be the total number of relevant documents for a query Q. If the document at rank k (D_k) is relevant, the precision p_k is calculated. Average Precision (AP) is the average of these precision values (Manning et al., 2008):

$$AP = \frac{1}{S} \sum_{k, D_k \text{ is relevant}} p_k. \tag{2.18}$$

MAP is the mean of APs over all queries. Geometric interpretation of MAP is the area under the uninterpolated precision-recall curve.

2.5.2 Metrics for Unknown-boundary Condition

The preceding metrics assume that the material is organized into a set of documents. The evaluation of spoken document retrieval in so-called *unknown-boundary condition*, where there is no topical segments or other

structure in the flow of audio, requires changing the metrics somewhat. Instead of returning a ranked list of documents, the system would return a ranked list of time pointers. In the TREC SDR (Garofolo et al., 2000) evaluations, the time pointers were mapped to the documents they fell in. Duplicate entries and pointers that fell between stories (commercials or fillers) were scored as non-relevant. This means that all topic boundaries in the evaluation corpus need to be known, which requires a lot of human resources.

An alternative is to use a "one-sided" metric, where the system returns a ranked list of replay points that should mark the onset of relevant content (Liu and Oard, 2006). The relevance assessment process consists of only finding the set of replay points that are relevant to the query. The closer in time the returned time pointer is to the human assessed ground truth point, the more score it is given. While knowing the end point of the segment would be useful for some applications like browsing, the user will notice when the topic changes and will know to stop listening.

A metric originally designed to be used with graded relevance judgements, *generalized average precision* (GAP) (Kekäläinen and Järvelin, 2002; Liu and Oard, 2006), can be applied to the case of evaluating ranked replay points. In this case, the relevance judgements are hard but the degree of match is graded. The degree of the match is determined by defining a relevance function R that takes maximum at the ground truth point g and slopes off as the returned replay point t moves further from g in either direction. The width w of the relevance function determines how far from g t has to move until R is set at 0 and the replay point is considered non-relevant.

For a ranked list of replay points, the precision at rank k is defined as:

$$p_k = \frac{1}{k} \sum_{i=1}^{k} R_i, \tag{2.19}$$

where R_i is the value of R for the replay point at rank i. The generalized average precision for a query with N ground truth points can then be calculated as

$$GAP = \frac{1}{N} \sum_{R_k \neq 0} p_k. \tag{2.20}$$

Values for R are calculated in the order of the list and each ground truth point is used only once. If a replay point is repeated nearby, it receives score of zero. Thus, systems that over-generate replay points are severely penalized. Figure 2.2 shows triangular relevance functions around two ground truth points.

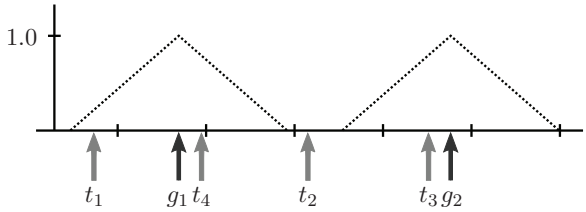


Figure 2.2. Relevance functions around two ground truth points, g_1 and g_2. Returned replay points $t_1, \ldots, t_4$ are processed in order: t_1 gets a low score, t_2 gets no score, t_3 gets a high score, and t_4 gets no score, because g_1 is already used by t_1.

Liu and Oard (2006) compare triangular, rectangular and Gaussian relevance functions and found that triangular functions are the most stable when different systems are ranked using GAP. Triangular functions receive the value of one at the ground truth point g and linearly decay until zero is reached at some distance w from g. 90 second wide windows are used for spontaneous Czech speech in the CLEF cross language speech retrieval track (Pecina et al., 2008).

3. Morphological Analysis of Words

Applications such as speech recognition, information retrieval and machine translation require that computers can process human languages. An important part of *natural language processing* (NLP) is building language models that capture the properties of the language. The complexity of the task of modeling a language is in part dependent on its *morphology*, that is, how words of the language are formed. For morphologically simple languages, such as English, it is often sufficient to use words as units of modeling. Morphologically rich languages are characterized by high degree of inflection, agglutination and compounding that may produce a very large number of word forms for a given root form. As a result, the language has a huge number of distinct word forms which makes using words as units of modeling inefficient. Using morphemes instead of words alleviates this problem significantly. In order to do so, we need a method that can analyze the morphological structure of any given word.

One solution is to use experts to form the list of words and linguistic rules needed to solve the morphology of word forms. These rule-based morphological analyzers exist for many languages. However, since the analyzer works on a limited set of words, some word forms can not be processed, such as rare words, dialect words or foreign words or names that enter the language. Another option is to learn morphological structure by observing regularities in language data. For example, the data may consist of example analyses for a subset of words, or it could merely be a collection of unannotated text. Therefore, there is no need for expert crafted labels or rules as long as there is data available for the language. This is especially important for less resourced languages that have no rule-based morphological analyzers available.

3.1 Morphology and Morphological Processing

A few linguistic concepts regarding morphology should be defined. *Morpheme* is the smallest meaning-bearing unit in language. A morpheme refers to the abstract meaning of a concept, while the *surface form* of the morpheme or *morph* is the realization of the morpheme in writing (or speech). For example, the word "unhappy" is composed of the morphs "un" and "happy" and the word "happier" is composed of the morphs "happi" and "er". The morphs "happy" and "happi" are *allomorphs*, different realizations of the same morpheme for the concept of happy. The reverse case, where a single form has more than one function, is called *syncretism*. For example, the suffix morpheme "s" can either mark plural of a noun or third person singular of a verb.

Inflection is the alteration of a word to express different grammatical categories, and *derivation* is the process of forming a new word on the basis of an existing word. Both can be achieved by joining *affixes* to *stems*. The process of concatenating morphemes to form words is called *agglutination*. Stem is the part of the word that is common to all of its inflected variants. The meaning of word *root* is related, but the difference is that derivational affixes are part of the stem, but not part of the word root. For example, "speaker" is a derivation from "speak", while "speaks" is an inflected form. The stems of the words are "speaker" and "speak", respectively, while both have the same root "speak". Sometimes different forms of a word have very different stems, for example the past tense of the word "go" is "went". However, both forms have the same dictionary form or *lemma*: "to go". Definitions of a *vocabulary* and *lexicon* vary, but in this thesis, a vocabulary refers to the list of words in the language, including all inflected forms, whereas a lexicon is a list of lemmas and other morphemes.

In some languages, all words are composed of a single morpheme, where in others words consist of several morphemes. The morpheme-per-word ratio defines where the language lies in the *isolating-synthetic* scale, a ratio of one meaning purely isolating language. A synthetic language is purely *agglutinative*, if the boundaries between morphs are clear, and it is *fusional*, if the morphs are overlaid in a way so that they are difficult to segment. The degree of fusion determines where the language falls in the agglutinative-fusional scale.

The desired output of morphological processing is different for differ-

ent applications, and thus different tasks can be defined. *Morphological segmentation* or *word decomposition* is the task of finding the boundaries between morphemes in words. This is a sufficient approach in applications such as speech recognition, where we are interested in modeling the sequences of surface forms of the morphemes. In information retrieval, morphological analysis is typically used for *lemmatization*, returning each word in text to its lemma. This is essentially a clustering task: the goal is to cluster all inflected forms of the same lemma to the same class so that documents are retrieved irrespective of which word forms are used to describe the contents. *Stemming* is a related approach that is especially used for IR in morphologically simple languages. Stemming algorithms use simple, heuristic rules to strip the ends of words so that different inflected forms are shortened to the same stem. While the Porter stemming algorithm (Porter, 1980) has been proven very effective for IR in English, simple stemming does not work as well for languages that are morphologically more complex. In full *morphological analysis*, all the morphemes in the words are identified. This task is the most difficult, because complex morphological phenomena such as allomorphy need to be taken into account.

3.2 Properties of the Finnish Language

Speech retrieval experiments in the publications of this thesis use Finnish as a test language. However, the results should extend to other languages with similar properties. A short description of the properties of Finnish is given in the following.

Finnish is a highly synthetic, agglutinative language. Suffixes are used for both inflection and derivation, and they can be strung together one after another. For example, the word “kirja” (a book) can take the inflected form “kirja+sta+ni+kin” (’from my book, too’). Derivatives of the word include “kirjanen” (a leaflet), “kirjasto” (a library) and “kirjain” (a letter). Finnish also frequently uses compounding for forming new words. Unlike many languages, Finnish allows component internal inflection: “kirja+n+lukija” (book+genitive+reader: ’reader of a book’). Some compounds are *transparent*, their meaning is consistent with the meaning of their components. For example, the words “sana” (a word) and “kirja” (a book) form the compound “sanakirja” (a dictionary). A compound is *opaque*, when its meaning can not be deduced by the meanings of the

components: the words “poika” (a boy) and “mies” (a man) combine to “poikamies” (a bachelor). For information retrieval, the amount of transparent and opaque compounds in the language has an effect when deciding whether compounds should be split to their components before indexing.

The complex morphological processes mean that Finnish has a very high number of unique word forms. It has been estimated that more than 2,000 word forms are possible for a single Finnish noun (Karlsson and Koskenniemi, 1985). This fact has an effect when designing Finnish NLP applications such as speech recognition or information retrieval.

Finnish has also fusional properties. In certain cases, the word stem changes when suffixes are attached. For example, the word “vesi” (water) has the genitive “vede+n”, partitive “vet+tä” and illative “vete+en”. This process complicates finding common stems of inflected forms for informational retrieval.

Homographs, words that share the same written form, but have a different meaning, is another consideration for IR. For Finnish, a special case is *inflectional homography* that happens when two or more different lemmas share an inflected form. For example, the word “alusta” can be one of seven different forms of five different lemmas: “alku” (a beginning), “alusta” (a base), “alustaa” (introduce), “alunen” (a coaster) or “alus” (a ship).

3.3 Rule-based Morphological Analysis

The relationship between the abstract (*deep-level* or *lexical-level*) representations of morphemes and the *surface-level* realizations can be expressed with a *generative grammar* that consists of an ordered sequence of context sensitive rewrite rules (Chomsky and Halle, 1968). Following the rules, an abstract representation is converted into a surface form through a series of intermediate representations. This approach is not well suited for analysis of surface forms to lexical-level, because while the rules are unambiguous from lexical to surface-level, they are ambiguous in the other direction, which means that one surface form can be generated in more than one way. For languages with a complex morphology, the computational complexity rises sharply.

In *two-level morphology* (TWOL) (Koskenniemi, 1983), a set of rules is applied in parallel (as opposed to sequentially) in the generative model.

The rules are context dependent symbol-to-symbol constraints that can refer to both lexical and surface-level representations at the same time. However, there are no intermediate levels of representation, hence the model is called "two-level". A lexicon of morphemes and words is used to restrict the search to allowed sequences of deep-level representations thus avoiding the ambiguity problem. Each rule and the lexicon of TWOL can be implemented as a *finite state transducer* (FST) and combined using intersection operation of the finite state algebra. Efficient FST algorithms can then be used for morphological analysis and generation.

3.4 Unsupervised Learning of Morphology

Designing the rules for the rule-based systems requires an amount of effort from people with linguistic training. An alternative is to observe statistical properties of text to learn a model of morphology. In *unsupervised learning*, the input data is unlabeled i.e. the desired morpheme segmentations or analyses are not available for any of the words. Unsupervised learning has the advantage that the resulting methods are language independent and models can be trained for any language with enough training text data available.

3.4.1 Morfessor

In this work, the morph-based approach for speech retrieval uses an unsupervised morphological segmentation algorithm called *Morfessor* (Creutz and Lagus, 2002; Creutz, 2007). The algorithm takes as an input raw, unlabeled text data and produces a model of morphology that can be used to segment words to morphs. The algorithm has a few variants, the one used in this thesis, *Morfessor Baseline*, is described in the following.

The first versions of Morfessor used *minimum description length* (MDL) formalism (Creutz and Lagus, 2002; Hirsimäki et al., 2006). The model can also be expressed in a Bayesian framework using a *maximum a posteriori probability* (MAP) estimation (Creutz and Lagus, 2005b, 2007). Both approaches aim to find a balance between modeling accuracy and model complexity, and produce equivalent results. The MDL formalism is used here.

Segmentation Model

Let us consider sending data x using an encoding model with parameters θ using the smallest possible number of bits. According to the MDL-principle (Rissanen, 1989), the optimal model is the one where the total code length of the model parameters $L(\theta)$ and the code length of data encoded with the model $L(x|\theta)$ is the shortest:

$$\underset{\theta}{\operatorname{argmin}}\, L(x, \theta) = \underset{\theta}{\operatorname{argmin}}[L(x|\theta) + L(\theta)]. \tag{3.1}$$

In morph segmentation, the encoded data is a text corpus. The model is the *morph lexicon*, a set of unique morphs. The morphs are strings of characters that are associated with their probability of occurrence. Each word in the corpus can be segmented using these morphs and the corpus can be represented as a sequence of pointers to the morph lexicon.

The length of the model representation is proportional to the size of the morph lexicon. The length of the corpus representation is proportional to the number of morphs in the corpus. At one extreme, the words are left unsegmented and the lexicon consists of all the individual words in the corpus. This leads to a compact representation of the corpus (a low number of pointers), but to a large representation of the model (a high number of unique morphs). At the other extreme, each word is segmented to individual letters. The model representation is compact, because the number of unique morphs is equal to the number of letters in the language. However, the corpus representation is very large because the number of pointers equals the number of letters in the whole corpus. Optimizing the model with respect to Equation 3.1 gives us a compromise between these extremes.

More formally, the code length for an individual character α is derived according to information theory from its probability by $L(\alpha)$=$-\log(P(\alpha))$. Using base 2 logarithms, the code lengths are measured in bits. The probability distribution for the letters in the language is assumed to be known. For a lexicon with M morphs μ_j, the code length of the lexicon is:

$$L(\text{lexicon}) = \sum_{j=1}^{M} \sum_{k=1}^{\text{length}(\mu_j)} -\log(P(\alpha_{jk})), \tag{3.2}$$

where $P(\alpha_{jk})$ is the probability for the k:th character in the j:th morph.

The probability distribution of the morphs in the corpus needs to be encoded also. The probabilities are estimated from the segmented corpus by counting the number of occurrences for each morph and the total number of morphs N. The probabilities can thus be encoded as integer values.

First, any positive integer N can be encoded using the following number of bits (Rissanen, 1989, p. 34):

$$L(N) \approx \log c + \log N + \log\log N + \log\log\log N + \cdots +, \tag{3.3}$$

where c is a normalization constant ($c \approx 2.865$). The morph frequencies can be encoded more efficiently as there are $\binom{N-1}{M-1}$ possibilities for choosing M positive integers that sum up to N (Rissanen, 1989, pp. 35–37):

$$L(\text{morph frequencies}) = \log \binom{N-1}{M-1}. \tag{3.4}$$

The entire model has now been encoded using $L(\theta) = L(\text{lexicon}) + L(N) + L(\text{morph frequencies})$ bits.

Now each morph μ in the corpus is encoded using $-\log(P(\mu|\theta))$ bits, where $P(\mu|\theta)$ is the probability of the morph. The length of the corpus encoded using the morph lexicon is:

$$L(x|\theta) = \sum_{i=1}^{N} -\log(P(\mu_i|\theta)), \tag{3.5}$$

where N is the number of morphs in the corpus, μ_i is the i:th morph. Since the code length of each morph has an inverse relationship to its probability, frequent morphs have shorter codes and rare morphs have longer codes. Thus, the optimization has the tendency to select frequent substrings as morphs and encode infrequent substrings as a sequence of morphs.

Search Algorithm

Equation 3.1 is minimized and the optimal morph segmentation found using a greedy search algorithm. At the beginning, all words are unsegmented. Each word is then processed in a random order. All possible segmentations into two parts are tried. If the best segmentation decreases the cost function (code length) compared to no segmentation, it is approved. The resulting two morphs are further segmented into parts recursively until no further improvement is gained. Every time a segmentation happens, it is applied to all words containing that morph. After all the words are processed once, the process is started over, but in a different random order. The algorithm stops when there is no significant reduction to the code length. Consult (Hirsimäki et al., 2006) for a detailed description of the search algorithm.

Using Morfessor for ASR and IR

After training, the morph segmentations for the words in the training corpus are known. However, these segmentations are not used as such.

Instead, Viterbi search is used to find the most probable segmentation for each word using the morph lexicon and the morph probabilities from the trained segmentation model. This way, all words can be segmented, including words that were not in the Morfessor training corpus. As will be further studied in Chapter 5, this is important especially for speech retrieval, because new words may be introduced at any time in queries.

When comparing the morphs produced by Morfessor to grammatical morphemes, it has been noted that if the word frequencies in the training corpus are ignored, and the model is trained on the word list rather than the corpus, the morphs match grammatical morphemes better (Creutz and Lagus, 2005b). If the word frequencies are used, and the training corpus is large, common word forms start to dominate the optimization and as a result the common words tend to be left undersegmented. Similarly, filtering out rare words, that are often misspellings and other noise, also improves the model.

Section 4.6 explains how Morfessor is used in speech recognition by training the language model on a text corpus segmented with Morfessor. In Finnish speech recognition experiments, the resulting morph-based language models have been found to perform better than word-based, syllable-based or models based on grammatical morpheme segmentations produced by a rule based morphological analyzer (Hirsimäki et al., 2006; Hirsimäki et al., 2009).

In text retrieval, Morfessor can be used instead of rule-based lemmatization or stemming. The words in the text corpus are segmented to morphs before indexing, as are the words in the queries. By matching morphs instead of whole words, different inflected forms of the same words can be matched if they share a common stem morph. In (Kurimo et al., 2010a), the performance of Morfessor for Finnish, German and English text retrieval tasks is compared. For all languages, the performance is clearly better than when using unsegmented words, but not as good as the best rule-based methods. However, for languages with no rule-based analyzers available, Morfessor could still provide advantages.

For speech retrieval, the morph-based approach has more advantages. Morphs can be used both for speech recognition and for retrieval. Compared to using word-based language models for Finnish speech retrieval, the morph-based approach is significantly better (Publication V). Using morphs for recognition, but traditional rule-based analyzers for indexing is also an option. However, recognition errors in the speech recognition

transcriptions may cause the analyzer to produce spurious results. The results indicate that morph-based and word-based indexing for Finnish speech retrieval produce about equal results, but combining the two approaches is better than either alone (Publication VI).

A small change to the Morfessor Baseline algorithm was made for Publication VI. The Viterbi search was modified so that instead of the most likely segmentation of a word, a number of alternative segmentations can be produced. The alternative segmentations are added to the query. This will help match the morphs in the recognition transcriptions better, because a word in different inflected form may be recognized using slightly different morphs.

Morfessor CatML and CatMAP

To reduce the over- and undersegmentation sometimes present in the Morfessor Baseline model, more sophisticated variants have been developed. In Morfessor Categories-ML (CatML) (Creutz and Lagus, 2004), the segmentation produced by the Baseline algorithm is reanalyzed using a maximum likelihood (ML) optimization. Morph usage patterns are used to tag each morph in the corpus as prefix, stem or suffix. Additional non-morpheme ("noise") category is used to tag morphs that do not fit to other categories. They are usually short segments resulting from oversegmentation of words. Heuristic rules are used to join together noise morphs to reduce oversegmentation and to split redundant morphs that consist of other morphs in the lexicon. A hidden Markov model, where categories are represented by states that emit morphs with particular probabilities, is used for assigning probabilities to each possible segmentation and tagging of a word form.

In Morfessor Categories-MAP (CatMAP) (Creutz and Lagus, 2005a), the heuristic join and split rules are replaced in favor of a hierarchical lexicon structure, where a morph can either consist of a string of letters or of two submorphs, which in turn can consist of submorphs. Prefix, stem, suffix and non-morpheme categories are again represented by HMMs and the model is now expressed in a maximum a posteriori (MAP) framework. The optimal level of segmentation is determined from the hierarchical representation by selecting the finest resolution that does not contain non-morphemes. In comparison to linguistic morphemes, CatMAP performs much better than Baseline, but in text retrieval experiments the performance of CatMAP is equal to Baseline for Finnish and German and

slightly worse for English (Kurimo et al., 2010a). In speech recognition, CatMAP has not been found to improve over Baseline (Creutz et al., 2007). So far, CatMAP has not been tested for speech retrieval.

4. Speech Recognition

An essential part of a speech retrieval system is speech recognition, the task of transcribing human speech into text. In this thesis, a research recognition system developed at Department of Information and Computer Science at Aalto University School of Science is used. Research on the subject has been ongoing for decades. Recent PhD theses on development of the speech recognition system include (Hirsimäki, 2009; Creutz, 2007; Siivola, 2006). The modern large vocabulary continuous speech recognition methods used by the system are described in this chapter.

The difficulty of a speech recognition task depends on the material that is transcribed. Background noise is one factor that makes speech recognition system performance degrade strongly. Also, the more the language in the speech differs from the language used in training the system, the more difficult the task is. Proper names and especially foreign names are difficult for recognition. Spontaneous speech (as opposed to planned speech) differs greatly both acoustically and linguistically from the speech and language typically used to train the recognizer. Spontaneous speech is characterized by non-grammatical sentences and disfluencies such as repetitions, fillers (“uh”, “well”), and false starts.

One likely use of speech retrieval methods is to enable searching of broadcast material. The range of speech types in broadcasts varies from planned speech in noise-free studio environment to spontaneous speech in noisy environments. There are applications that use even more difficult material, for example retrieval on recordings of lectures, meetings or telephone conversations. Therefore, it is to be expected that the recognition performance is far from perfect. Further complication from retrieval point of view is that even if the overall performance of the recognizer is good, missing a word that is important for retrieval, such as a proper name, can cause a relevant segment to be missed in retrieval.

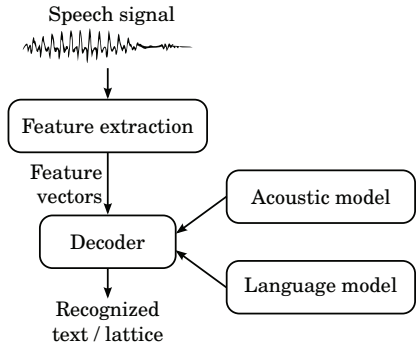


Figure 4.1. Overview of the speech recognition system.

4.1 Overview

In the probabilistic framework, speech recognition can be viewed as the task of finding the most likely sequence of words $\hat{W}$, given the observed speech signal O:

$$\hat{W} = \underset{W}{\operatorname{argmax}} P(W|O). \tag{4.1}$$

Using the Bayes rule and noting that $P(O)$ remains constant with respect to W, the maximization can be written as:

$$\hat{W} = \underset{W}{\operatorname{argmax}} \frac{P(W)P(O|W)}{P(O)} = \underset{W}{\operatorname{argmax}} P(W)P(O|W). \tag{4.2}$$

The probabilities $P(W)$ are given by the *language model* (LM) and the probabilities $P(O|W)$ by the *acoustic model* (AM). The observation O is given as a sequence of *feature vectors*. In theory, the maximization process would involve looking through all possible sequences of words and choosing the most probable one. That is the task of the *decoder*. In practice, the search space will have to be limited. Figure 4.1 illustrates the speech recognition process.

4.2 Feature Extraction

Digitized speech signal comprises of a sequence of numbers that represent the amplitude of the signal measured at certain intervals. The *sampling rate* defines the number of samples taken by second and the *resolution* defines the number of bits used to store each sample. Typically, in speech recognition, 16kHz (16,000 samples per second) 16-bit audio is used for input. The signal at this form is largely redundant and the dimensionality needs to be reduced before further processing. *Feature extraction* aims to extract features that capture the information of the phonetic structure

while discarding the rest.

The recognizer used in this thesis uses *Mel-frequency cepstral coefficients* (MFCC) (Davis and Mermelstein, 1980) for this purpose. First, the signal is divided into frames by taking a 25ms Hamming window at 8ms intervals. Fast Fourier transform (FFT) is used to calculate the short time power spectrum of each frame. The power spectrum is divided into filter banks using triangular overlapping windows and the logarithm of the energy in each filter is accumulated. Rather than using filters that are spaced evenly in the frequency axis, the frequency is first transformed to perceptually motivated frequency scale called the *Mel-scale*. On the linear scale, this corresponds to narrow filters at low frequencies and wider filters at high frequencies. Finally, discrete cosine transform (DCT) is used to produce the 12 dimensional MFCC feature vector. The feature vector is extended with the average power of the frame. In cepstral mean subtraction (CMS) (Atal, 1974), the mean of the 150 surrounding frames is subtracted from each vector to reduce the effects of convolutional distortions in the channel. The feature vector is appended with so called *delta* and *delta-delta* components, that is, the first and second time derivatives of the 13 components resulting in a 39-dimensional vector. The mean and variance are normalized globally with a linear transform. Finally, another linear transform, maximum likelihood linear transform (MLLT) (Gales, 1998), is used to reduce the effect of speaker and environment variation.

4.3 Acoustic Modeling

The acoustic model likelihoods $P(O|W)$ are obtained using hidden Markov models (HMMs). In *monophone* modeling, an HMM is trained for each *phoneme* of the language. Phonemes are the smallest distinctive units of sounds in speech. However, depending on the context, a phoneme may be pronounced in different ways. The realizations of phonemes in speech are called *phones*, and the variant phones that correspond to the same phoneme are called *allophones*. In speech recognition, context-dependent models are used for this reason. A different model is trained for each *triphone*, that is, the phoneme and its neighboring phonemes. The HMMs have a number of parameters Λ that need to be estimated from training data. Most important are the emission distributions, that are modeled by *Gaussian mixture models* (GMMs) with diagonal covariance matrixes. Since triphones can be similar to each other and since there is

usually not enough training data to model every possible triphone, some triphones are clustered together. Decision-tree based clustering was used to merge HMM states with similar distributions (Young et al., 1994). The HMM based approach for speech recognition is thoroughly explained in (Rabiner, 1989).

A *pronunciation dictionary* is used to map the words in the language model to the phonemes represented by the acoustic models. Finnish orthography (the relationship between phonemes and graphemes) is simple: each letter stands for one sound and each sound is represented by the same letter. The exception is the sound /ŋ/, which is written as n in the short form (kenkä: [keŋkæ]) and as ng in the long form (kengän: [keŋŋæn]). Since the phoneme models are context dependent, /ŋ/ is not modeled explicitly but as different variants of n and g. Letters that are not native to Finnish but appear in loan words and foreign names (c, q, w, x, z, å) are transformed to Finnish phoneme labels by the simple rules: ch $\rightarrow$ ts, c $\rightarrow$ k, qu $\rightarrow$ kv, w $\rightarrow$ v, x $\rightarrow$ ks, z $\rightarrow$ ts, å $\rightarrow$ o.

In acoustic model training, an objective function that is dependent on the training data is maximized. Maximum likelihood (ML) training uses the likelihood of the observations in the training data given the HMM of the reference transcription S_r as the objective:

$$F_{ML}(\Lambda) = P(O|S_r, \Lambda). \tag{4.3}$$

In discriminative training, the objective function measures the recognition accuracy instead of likelihood of the data. Using minimum phone frame error (MPFE) criterion, the objective function is (Zheng and Stolcke, 2005):

$$F_{MPFE}(\Lambda) = \sum_S P_k(S|O, \Lambda)A(S, S_r), \tag{4.4}$$

where $P_k(S|O, \Lambda)$ is the posterior probability of hypothesis S, k is an acoustic scaling parameter, and $A(S, S_r)$ is a measure of number of frames having the correct phone label.

In Finnish, phone duration is sometimes the only clue for discriminating between certain words. Duration is marked orthographically by one versus two letters: “takka” (a fireplace), “taakka” (a burden), “takaa” (from behind). Measured by difference in letters, an error in phone duration may not seem significant, but for IR it can make a difference. HMMs model phone durations rather poorly, therefore explicit phone duration for Finnish speech recognition have been tested (Pylkkönen and Kurimo, 2004). A rescoring of recognition hypotheses according to a gamma dis-

tribution that models the state durations was found to produce a good accuracy while keeping the process efficient.

4.4 Language Modeling

The probability of the word sequence $P(W)$ in Equation 4.2 is assigned by a language model. Also, the language model is used to restrict the search space to sequences of phonemes that correspond to actual words in the language. The recognition system used in this thesis, as almost all modern speech recognition systems, uses *n-gram* models for statistical language modeling. If W is a sequence of m words $w_1, \ldots, w_m$, $P(W)$ can be factored as:

$$P(W) = \prod_{i=1}^{m} P(w_i | w_1, \ldots w_{i-1}). \tag{4.5}$$

In an n-gram model, the conditional probability $P(w_i|w_1, \ldots w_{i-1})$ is approximated by taking the *n:th order Markov assumption*, that is, assuming that the probability is only dependent on previous $n-1$ words:

$$P(W) \approx \prod_{i=1}^{m} P(w_i | w_{i-(n-1)}, \ldots w_{i-1}). \tag{4.6}$$

For example, a trigram model would only take into account the two previous words. A training corpus is used to estimate the probabilities by counting the number of occurrences of every n-gram (the sequence of n words) in the corpus:

$$P(w_i | w_{i-(n-1)}, \ldots w_{i-1}) = \frac{C(w_{i-(n-1)}, \ldots w_{i-1}, w_i)}{C(w_{i-(n-1)}, \ldots w_{i-1})}, \tag{4.7}$$

where $C(x)$ is the number of times x appears in the training data. While this is the optimal model in terms of *maximum likelihood*, in practice it will work well only for the frequent n-grams, because it will over-learn the training data and assign a zero probability to all n-grams that do not appear in the training data.

Smoothing methods are used to modify the probability estimates so that some of the probability mass is moved to the n-grams that are underestimated. Several smoothing methods have been proposed, and a comprehensive evaluation of methods is given in (Chen and Goodman, 1999). In *back-off* methods, if an n-gram receives zero probability, lower order estimates are used instead (Katz, 1987). In *interpolation* methods, lower order estimates are also used, but in this case, by interpolating all estimates with lower order estimates down to 1-grams (Jelinek and Mercer, 1980).

The interpolation strategy means that some high-order n-grams can be removed from the language model, if the low-order estimates are good enough. This may be necessary in order to limit the language model size. The language models in this thesis use Kneser-Ney smoothing (Kneser and Ney, 1995) and are fixed length n-grams in Publications III and IV and variable length n-grams built with a growing and pruning algorithm (Siivola et al., 2007) in Publications V and VI.

4.5 Decoding

Given the observations O, acoustic model likelihoods $P(O|W)$, the pronunciation dictionary and the language model probabilities $P(W)$, the process of maximizing Equation 4.2 is called *decoding*. Due to the huge number of combinations of vocabulary words, an exhaustive search of all possible word sequences is intractable. The decoder must limit the search to the word sequences that are deemed most probable. In this work, a decoder (Pylkkönen, 2005) based on the token-passing paradigm (Young et al., 1989) is used. The search is performed by moving tokens in a cyclic search network containing the HMM state sequences for all the words in the vocabulary. To restrict the search space, only transitions that correspond to word sequences in the language model are allowed. Each token corresponds to a hypothesis and it is updated at each frame:

1. The token is propagated through the transitions leaving from its current state.

2. The acoustic probability is updated.

3. If the token arrives at a state that defines the next word, the LM probability is also updated.

4. Tokens that share the same state and LM history are merged so that only the token with the highest probability is preserved.

After all tokens have been updated, low probability tokens are removed to control the speed of the decoding. *Beam pruning* means keeping only the tokens that have their probability higher than the highest probability minus a constant beam value. In *histogram pruning*, only a constant

number of tokens are retained after each step.

The decoder represents the search space of hypotheses it explores as a *lattice*, that can also be extracted and saved for further use. A lattice is a directed acyclic graph in which each node represents a point in time and an arc between nodes indicates that a word occurs between the times of the start and end node. Each arc is associated with acoustic and language model likelihoods. The 1-best hypothesis is the path through the lattice that has the highest probability. Less likely paths through the lattice can also be found and represented as an *N-best list*, that contains the N most likely paths through the lattice and their probabilities.

Lattices also enable *multi-pass* search strategies, where in the first pass, a lattice is produced using simple acoustic and language models, and in the subsequent passes, the resulting lattices are rescored using more complex models (Richardson et al., 1995). This allows using models in the later passes that would be too complex in the first pass. However, if the right alternative is pruned in the first pass, it can never be recovered.

An alternative decoding approach is using weighted finite-state transducers (WFSTs), in which separate transducers are defined for all models (acoustic models, language models, pronunciation dictionary) (Mohri et al., 2002) that are then combined to a single transducer. The approach allows very fast decoding, but the transducer composition is memory intensive.

4.6 Morph-based Speech Recognition

The preceding sections of this chapter assumed that the language model is based on sequences of words. This is a suitable approach for languages such as English for which a reasonable set of words can cover commonly occurring usage of the language. For languages such as Finnish, Turkish and Estonian, inflection and compounding causes the number of distinct word forms to become huge. Many concepts that take several words in English can be expressed by using a single word. For these languages, using words as language modeling units becomes inefficient for two main reasons. First, a large vocabulary expands the search space of the recognizer and the time and memory requirements for the search process grow large. Second, a large number of words leads to *data sparsity* problems, because if the corpus does not contain enough instances of each word in each context, the estimates of n-gram probabilities are not reliable.

The recognizer vocabulary is typically limited to a number of most frequent words in the language model training text corpus. The words not in this set, the *out-of-vocabulary* (OOV) words, are replaced with a special symbol and are thus not modeled. Using an independent test set, it is possible to calculate the so called OOV-rate to measure the coverage of the vocabulary. For English, a vocabulary of about 65,000 will result in an OOV-rate of 0.31%–0.65% (Woodland et al., 1995). For Finnish, a 69,000 word vocabulary will have 13.1%–19.9% OOV-rate (Hirsimäki et al., 2006). Even increasing the vocabulary size to 500,000 words, the OOV-rate will remain at 5.4% (Hirsimäki et al., 2009). A vocabulary of this size will significantly increase recognition time and memory requirements. Any word in speech that is not in the vocabulary will be misrecognized and substituted by a phonetically similar word. Further, for every wrong word substituted or inserted, the language model probabilities for the following words will be unreliable and the recognizer can produce other errors. On average, every OOV word in speech will cause 1.46 erroneous words in English ASR (Hetherington, 1995).

An alternative is to use suitable subwords as language modeling units. The shorter the subwords, the smaller the vocabulary needed to cover the language, but on the other hand, the larger the number of subwords in the corpus. The Morfessor algorithm (Section 3.4.1) can be used to segment the words in a text corpus to morphs. The language model can then be trained on the segmented corpus and the resulting n-gram will learn to model the structure of words as well as the structure of sentences. Morfessor can segment every word it encounters (in the extreme it will split it to individual letters) and the resulting language model will be able to model even the most infrequent words at least to some extent. Smoothing the language model will ensure that every possible combination of morphs will receive a positive probability and, at least in theory, it is possible to recognize word forms that do not appear in the training corpus at all.

All language modeling methods presented in this chapter can be applied to morph language models with minor changes. First, the pronunciation of each morph needs to be defined. For Finnish this is easy since the almost one-to-one mapping of graphemes and phonemes means that the pronunciation of any subword string stays the same irrespective of its context. For languages with more complex orthography such as English, the different morph variants with different pronunciations can be made unique by numbering (Creutz et al., 2007). Second, since we want to tran-

scribe the speech in words, the recognizer will need to be able to assign word breaks as well. This is achieved by introducing a special word break symbol into the training text corpus. The symbol is processed like any other token in text and the recognizer will learn to produce the symbol at word boundaries. Third, the n-gram order needs to be extended. A fixed number of units will span shorter amount of text, if the size of the unit is reduced. The word break symbols further increase the number of units to be modeled. Therefore, to use the same amount of information in the linguistic context, the recognizer will need to use higher order n-grams (Hirsimäki et al., 2009).

A review of speech recognition experiments for agglutinative languages using morph and other subword-based approaches is given in (Hirsimäki et al., 2009). An alternative to the statistical morphs produced by Morfessor is to use rule-based morphological analyzers to segment the words to grammatical morphemes. The drawback is that the morphological analyzers work on a limited vocabulary and can not process all words. In Finnish ASR, the statistical and grammatical approaches produce about equal error rates, both performing significantly better than word-based LMs, even with very large 500k word vocabularies (Hirsimäki et al., 2006; Hirsimäki et al., 2009). Similar results have been achieved for Estonian speech recognition (Puurula and Kurimo, 2007; Hirsimäki et al., 2009). For Turkish, word-based approach has been reported better when using 2-gram models (Arisoy et al., 2006), but for longer, up to 6-grams models, morphs perform better (Kurimo et al., 2006b). Arabic experiments show that morph-based approach is better when compared to 64k word vocabularies (Xiang et al., 2006; Choueiter et al., 2006), or 256k word vocabulary (El-Desoky et al., 2009). If the size of the word vocabulary is increased to 300k (Xiang et al., 2006) or to 800k (Choueiter et al., 2006), word and morph-based LMs perform at equal level. El-Desoky Mousa et al. (2010, 2011) use Morfessor for German LVCSR and report improvements over standard word-based models, and also over subword models based on supervised decomposition.

4.7 Evaluation Metrics

To evaluate speech recognition performance, the transcription produced by the recognizer is compared to ground truth reference transcriptions. The minimum number of insertions (I), deletions (D) and substitutions

(S) needed to transform the reference to the recognition output is counted. The counts are used to compute the most commonly used metric for speech recognition, the *word error rate* (WER):

$$\mathrm{WER} = \frac{I + D + S}{W} \cdot 100\%, \tag{4.8}$$

where W is the total number of words in the transcription.

WER is not comparable across languages, because the same information can be expressed in one language using fewer words than in others. If a word is composed of several morphemes, an error made with one morpheme will cause the whole word as counted wrong. An alternative is to calculate the *phoneme error rate* (PER) or *letter error rate* (LER), where the phoneme or letter insertions, deletions and substitutions are counted. For Finnish that has a simple orthography, PER and LER are very similar. The LER reflects the amount of labor needed to manually correct the recognizer output and is more comparable across languages.

From speech retrieval point of view, not all errors are equally important. *Function words*, such as "the", "is", "at", "which", etc. are usually filtered out before indexing by using a *stop list*. Any recognition error made with these words has no effect on the resulting IR performance. The effect of inflection is removed from the recognizer output by using stemming or morphological analysis methods (Section 5.2.2). Recognition errors that change the inflection of a word are thus neutral from IR point of view. Also irrelevant is the word order, because most IR models do not take it into account. However, a substitution error causes the correct word to be replaced by an incorrect one. For IR this may be twice as damaging as deletion or insertion. For these reasons, another metric, *term error rate* (TER), has been suggested for evaluating transcription errors based on the number of errors that effect the IR performance (Johnson et al., 1999):

$$\mathrm{TER} = \frac{\sum_w |R(w) - T(w)|}{W} \cdot 100\%, \tag{4.9}$$

where $R(w)$ and $T(w)$ are the number of times word w appears in the (stopped and stemmed or lemmatized) reference R and the transcription T.

In a speech retrieval application, possibly thousands of hours of speech need to be processed. It is therefore important to consider the processing time the speech recognition takes. Usually, processing time is expressed in terms of *real-time factor* (RT-factor), that is, how many times the length of the audio the processing takes. Since the amount of spoken material is

vastly expanding, at most RT-factor of one, that is, real-time processing, is often desired. The processing time can be reduced by increasing pruning in the decoder, but there is a trade-off between processing time and recognition accuracy. When comparing different systems, their RT-factors should be around the same for fair comparisons.

5. Morph-based Speech Retrieval

Systems for retrieving spoken content typically consist of three parts. First, the speech is transformed into textual form with a speech recognizer. Second, the recognized text is indexed with an information retrieval system. Third, the user inputs query terms that are matched to the terms in the index and the portions of speech that best match the query are returned. Each of these steps have potential sources of error that can cause retrieval performance to suffer.

The first and the biggest source of errors is the ASR. On suitable material, that is, planned speech recorded in clean conditions such as TV news anchor reading aloud text, the word error rates can be less than 20%. At error levels this low, the retrieval performance is indistinguishable from retrieval using error free human transcriptions (Garofolo et al., 2000). But if the speech is spontaneous or the recording conditions are noisy, the error rates will sharply rise. Many errors are caused by the limited vocabulary of the recognizer. From retrieval point of view, the limited vocabulary is especially problematic. Query words are often low frequency words, because they are selected to be discriminative. Thus it is likely that a query word is an OOV word or a word with unreliable language model estimates due to data sparseness. As seen in Chapter 4, the morph-based approach for speech recognition can help with the OOV problem especially for agglutinative languages. Any word in speech can potentially be transcribed by recognizing its component morphs.

Another source of errors arises from the properties of the language of the speech. Words at different locations will appear in different inflected forms. Before indexing, the effect of inflection should be normalized. Traditional approach is to use stemming or lemmatization to transform inflected forms to a common base form. Using rule-based analyzers, the OOV-problem is encountered again, since the analyzers also work on a

limited lexicon. Recognition errors cause further complications for the analyzers. However, if morphs are used as index terms as such, the need for further morphological analysis is avoided.

A related problem is the difference in words that are used to describe the same concept in the query and in the document. This synonymy problem can be alleviated by adding words to the query that have similar meaning to the original query words (query expansion) or by conflating words with similar meaning before indexing. The methods can either use linguistic knowledge, such as a thesaurus, or try to learn from data which words have similar meanings by observing co-occurring words. In the morph-based indexing, the stems of different inflected forms can be different due to non-concatenative morphological processes. This is a problem that is similar to synonymy. Section 5.2.3 explains how latent semantic indexing and alternative morph segmentations are used in the morph-based system to deal with the problem of unoptimal stem morphs as well as the problem of synonyms.

5.1 Subword Units for Speech Retrieval

In addition to the morph-based approach presented in this thesis, a number of subword-based approaches have been proposed for speech retrieval. In theory, it is possible to transcribe speech as a sequence of phonemes and use the phonetic transcription as the basis of retrieval. Query words are transformed to phoneme sequences as well using text-to-speech technology and the phoneme representation of the query is matched to the phonetic transcriptions. Since any word in speech can be expressed in phonemes, the OOV problem is completely avoided. However, since language models based on phoneme strings are less capable of modeling the structure of the language, the error rates of phonetic speech recognizers are much worse than the error rates of word-based systems. The high error rates severely degrade the possible retrieval performance as well.

Ng and Zue (2000) investigate extracting phone sequences from phonetic recognizer transcription and using them as indexing units for spoken document retrieval in English. They compare phone n-grams of different length, syllable-like units and *multigrams*, non-overlapping, variable-length, phonetic sequences. The multigrams are discovered using an unsupervised learning algorithm that finds the segmentation into phoneme substrings that maximizes the likelihood of the data (Deligne and Bimbot,

1997). Phone 3-grams are found to work best, but compared to retrieval from error-free transcriptions, the performance falls by 60%. The errors caused by the phonetic recognizer can be somewhat compensated by using robust retrieval methods. For example, by estimating an *error confusion matrix* of how probable it is that a certain phone is recognized as a certain other phone, it is possible to expand the query with errorful variants of the original terms with appropriate weights.

In this work, rather than extracting subwords from phonetic transcriptions, subwords are used already at the recognition phase by using a subword language model. Similar approach is adopted in Olsson (2008), where phone multigrams, trained on the phonetic representation of a text corpus, are used for language modeling. The performance for spoken utterance retrieval in English was at approximately same level as using a word-based system, but significantly improved compared to a phone-based approach. Logan et al. (2002) also use phone strings, called *particles*, for language modeling and retrieval. Particles are syllable-like phoneme strings that are determined by maximizing the leaving out likelihood of a particle 2-gram language model (Whittaker et al., 2001). Compared to a word-based system for English speech retrieval, the particle-based system yields better MAP for OOV queries, but the word-based system performs better overall. Best results are achieved by combining word and particle-based results.

Another data driven subword language modeling approach is based on *graphones* (Bisani and Ney, 2005). Originally designed for grapheme-to-phoneme conversion in text-to-speech systems, a graphone is a pair of letter and phoneme sequences of possibly different length. Both the orthographic form and the pronunciation of a word are regarded as being generated by a sequence of graphones. Graphones are learnt from a pronunciation dictionary by a data driven algorithm. Akbacak et al. (2008) use graphones for English STD. A flat hybrid word-graphone language model is trained: words that would otherwise be left OOV are replaced by their graphone decomposition in the language model training corpus. After recognition, all graphone sequences in the transcripts are joined into words. STD performance is improved for both IV and OOV words.

Syllables are a natural unit for language modeling and retrieval for syllable-based languages and have been successfully used e.g. for Mandarin Chinese (Chen et al., 2002). The approach is less intuitive for other languages, but has still the potential to reduce the effect of OOV words.

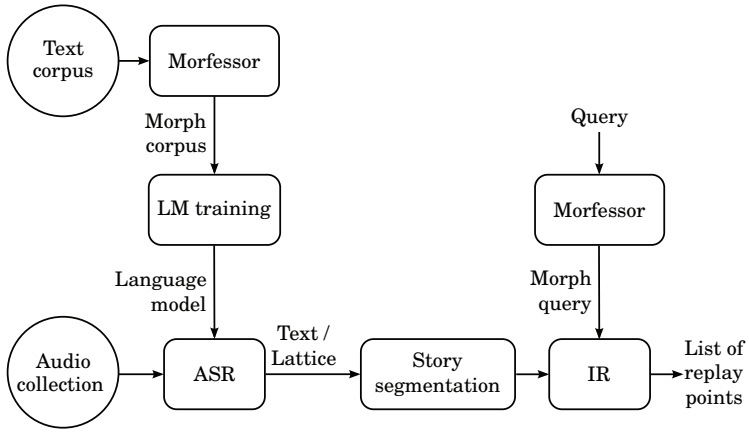


Figure 5.1. Overview of the morph-based retrieval system. First, Morfessor is trained using a text corpus. The same Morfessor model is then used to segment both the language modeling training corpus and the query.

Larson et al. (2007) test syllables as language modeling and indexing units for German speech retrieval. Syllable 2-grams are found the most effective as indexing features for both ASR transcriptions and for error-free text, but transcriptions produced with word-based language models work best. Fuzzy match based on the edit distance between query syllables and syllables in transcriptions are found to improve retrieval performance. Compared to morphemes, syllables seem less intuitive as indexing units, because they are not associated with a unique meaning. In Finnish speech recognition experiments, syllable language models produce bigger error rates than morph-based language models (Siivola et al., 2003).

5.2 Morph-based Retrieval

This thesis proposes a morph-based approach for speech retrieval in agglutinative languages such as Finnish. In the morph-based system, subwords discovered with the unsupervised Morfessor algorithm are used both as language modeling and information retrieval units. The morph-based speech recognizer transcribes the speech as a sequence of morphs with word boundaries marked with a special symbol. The most straightforward approach is to use morphs as index terms as such, but methods using words or combination of morphs and words are also possible. Overview of the morph-based approach used in this thesis is given in Figure 5.1.

The resulting retrieval performance of the morph-based system is com-

Table 5.1. Example recognition results of two unseen query words at two different locations each. With the morph LM, it is possible to recognize correctly at least some of the morphs, which will match morphs in the segmented query. With the word LM, the words are replaced by similarly sounding but unrelated words. The name "Iliescu" was not in the morphological analyzer lexicon and was thus left unprocessed for the word query.

Query word	Iliescun		Namibian	
- Translation	*Iliescu's*		*Namibia's*	
Morph query	ili escu n		na mi bi an	
Morph LM rec.	n ilja escu	ili a s kun	ami bi an	na min pi an
Word query	iliescun		namibia	
Word LM rec.	lieskoja	eli eskon	anjan	namin pian
Word lemmas	lieska	eli elää esko	anja	nami pian pia
- Translation	*flame*	*or live Esko*	*Anja*	*candy soon Pia*

pared to different baseline approaches. The baseline ASR uses a word-based language model with a very large vocabulary (Publication V) and the baseline information retrieval approach uses a rule-based morphological analyzer for lemmatization (Publication VI). The effect of OOV query words for the different approaches is studied further in a retrieval scenario where the query OOV-rate is artificially increased (Publication V).

5.2.1 Selection of Language Modeling Units

In Publication V, morph-based and word-based language models were compared in terms of resulting recognition and retrieval performance. For the word language model, about 490,000 of the most common word forms in the training corpus were included, leaving about 4.7% of the words in the training set OOV, but still keeping the vocabulary comparatively large. The morph language model had only about 19,000 morphs in the lexicon, but since any word can be generated by a concatenation of morphs, the OOV-rate is effectively 0%. In the word-based approach, a rule-based morphological analyzer (Koskenniemi, 1983) was used to transform each word in the transcriptions and in the queries to its lemma (base form). In the morph-based approach, morphs were used as index terms as such and Morfessor was used to segment the words in queries to morphs. The experiments were performed on the small Tampere-corpus.

Some of the least frequent words in the training corpus had to be excluded to limit the size of the resulting word LM. If an OOV query word (Q-OOV) is used, the resulting retrieval performance will severely suffer.

Table 5.2. Morph-based and word-based LMs compared for recognition and retrieval in normal (N) and high Q-OOV (H) scenarios.

	morph (N)	morph (H)	word (N)	word (H)
WER (%)	26.01	29.62	28.63	34.74
LER (%)	7.50	8.50	8.30	10.10
RT-factor	1.23	1.27	2.17	2.34
MAP (%)	84.4	64.2	77.9	48.0

In the morph-based approach there are no OOVs, because any word can be expressed as a sequence of morphs, but it is still possible that a query word did not appear in the training corpus at all. Morfessor can segment even these so-called "unseen" query words to morphs and it is possible to recognize the same word in speech as a sequence of morphs (Hirsimäki et al., 2009). To study the effect of unseen words, a few descriptive words were selected from each query and an another version of the training corpus was constructed so that any sentence with any of these words in any inflected form was excluded. The difference in performance of word and morph LMs were compared in this so-called "high Q-OOV" scenario. Table 5.1 shows an example how unseen words are recognized with morph and word language models and how the words are processed for queries in each case.

A summary of the results is presented in Table 5.2. The morph-based approach is better than the word-based both in terms of transcription error-rates and the resulting retrieval performance. Especially in the high Q-OOV case, the performance of the word-based system is severely degraded, as hypothesised, even though the RT-factor of the morph-based system is significantly better. See Publication V for details of the experimental setup and results.

5.2.2 Selection of Indexing Units

The preceding section confirms that morphs are superior to words as language modeling units for Finnish SDR, but whether it is best to use them as indexing units as well requires further comparisons. The rest of the experiments all use morph LMs for speech recognition, but vary in the way the resulting transcriptions are used for indexing. The options include:

1. Use morphs as index terms and use Morfessor to segment the query words.

2. Join morphs to words and use a morphological analyzer to transform each word in the transcriptions and in the queries to its base form.

3. Combine the morph and base form transcriptions, and use both forms in the index and in the queries.

4. Query the morph and base form indexes separately, and combine the ranked lists.

The first choice is attractive because the need for a morphological analyzer is avoided. The drawback is that sometimes the morphs produced by Morfessor are different for different inflected forms due to non-optimal segmentation or due to the fact that the word root may change for different inflected forms.

The standard in IR of agglutinative languages is to use a morphological analyzer for finding base forms. A disadvantage of this approach is that the analyzer works on a limited lexicon and any word not in the lexicon will have to be left unprocessed. Interesting words from IR point of view, such as names of people, may not be returned to their base form. In the case of transcriptions produced by the morph recognizer, there is another downside. If one of the morphs in the word is misrecognized, the word may change into an ungrammatical form not recognized by the analyzer or into a form of an unrelated lemma. For example, the name "Eero Heinäluoma" was in one instance recognized as morphs "vir heinä <w> luoma", which roughly translates to "created as mistakes". The words are normalized by the morphological analyzer to base forms "virhe" (mistake) and "luoda" (to create). Using morphs as index terms, the query morphs "heinä"' and "luoma" could still be matched in the transcriptions.

Combining the preceding two approaches has obvious advantages. The word forms that the morphological analyzer can successfully process are returned to a common base form and the partially correct words can still be matched by their component morphs. In the simplest, the morph and base form transcriptions are concatenated into a single transcription and then indexed. However, since the resulting transcription has both base form and morph terms, that sometimes share the same form and sometimes not, it is possible that the resulting term weights will be unoptimal. Another option is to construct and query the morph and base form indexes separately and combine the resulting ranked lists. A simple approach of

Table 5.3. Retrieval results comparing different indexing units. Results on the Tampere-corpus (Tam.) and the text corpus are mean average precisions (MAP) and results on the Podcast-corpus (Pod.) are mean generalized average precisions (MGAP). The Podcast corpus has been tested using long and short versions of the queries. Podcast corpus has been tested on normal (N) and high Q-OOV scenarios (H). The results on the latter scenario are previously unpublished.

Corpus	Tam.	Tam.	Pod. (N)		Pod. (H)	
Ref.	(PII)	(PIII)	(PVI)		-	
WER	30.4	34.0	-		-	
LER	7.1	11.2	-		-	
num. morphs	65k	26k	19k		19k	
Query	long	long	long	short	long	short
morph	79.2	77.7	43.0	30.4	34.2	19.0
base form	78.0	75.7	43.8	34.7	36.9	21.3
comb.	87.5[1]	-	46.0	37.3	40.3	26.1
interl.	-	-	46.6	35.1	38.9	22.6

interleaving was adopted here: the final ranked list was constructed by picking items in order, alternating between the two lists, and removing duplicates. The former approach is called the *combined* method and the latter the *interleaved* method.

The choice of indexing units for speech retrieval has been tested in Publications III and VI of this thesis, and also previously in (Kurimo et al., 2005). Selected results have been collected in Table 5.3. There have been some changes in the recognizer and retrieval system setups between publications, which makes comparisons slightly more difficult and the results are indeed somewhat mixed. One factor that has changed is the Morfessor and language model training corpus and the resulting morph lexicon size, which can cause variance in the morph-based results. Also the morphological analyzer was updated over the years, and its use perfected, improving the quality of the base form index.

On the Podcast corpus, experiments were run using two different versions of the queries: long and short. The results indicate, that for the speech corpora, morph and base form indexes yield about equal performance when using long queries. If the query is short, the base form index is somewhat better. This is explained by the fact that if a query word gets segmented to morphs in an unoptimal way, the performance suffers. For short queries, unoptimal segmentation of just one word can be detrimental, but for long queries, there are other words that can compensate.

[1](Kurimo et al., 2005)

However, combining morph and base form methods always leads to a level of performance that is better than either approaches alone.

In retrieval experiments on a text corpus (Kurimo et al., 2010a), base forms perform better than morphs, which indicates that, in the absence of recognition errors, the morphological analyzer can find good index terms more reliably. However, if the Q-OOV-rate is increased for the Podcast-corpus, the amount of errors increase, and the base form approach gets comparatively better for long queries and comparatively worse for short queries. The long queries have more words that are not 'unseen', and if these words can be reliably recognized and turned into base forms, they work well and can have a large effect in retrieval. However, in the short queries, a much larger proportion of words are previously unseen. For these words, it seems that morphs work better as index terms, and the difference in performance decreases. Finally, when combining the indexes, the increase in MAP in the high Q-OOV scenario is greater than in any other case, clearly indicating that morphs as index terms do provide additional useful information.

5.2.3 Reducing the Effect of Allomorphy on Speech Retrieval

The morph-based indexing approach suffers from the fact that the statistically determined morphs boundaries do not always fall on optimal locations. The Morfessor algorithm will choose frequent substrings as morphs and split infrequent substrings into smaller units. This may cause common inflected forms of a word to become undersegmented and rare forms to become oversegmented. Further, the same morpheme may be realized in different surface forms. Most importantly, due to linguistic phenomena such as consonant gradation, the spelling of the word root may change when a suffix is added. Thus, the different inflected forms may not share a common stem that other words do not also share.

The problem of differing word stems resembles the problems of synonymy. An information retrieval method developed to solve the problem of synonymy is latent semantic indexing (LSI) (Section 2.3.4). Applied to morph-based indexing, LSI can potentially project different morphs with the same meaning to the same dimension. Words that are important to the topic will appear in the document many times and often in different inflected forms. If the different inflected forms produce different stem morphs, LSI can infer by co-occurrence statistics that the morphs have related meaning.

LSI was applied to morph-based indexing in Publication III using 150 latent dimensions. The MAP was improved from 77.7% to 82.7% for the morph index and from 75.7% to 82.9% for the base form index. Thus, relative improvements were bigger for the base form index, which however started at a lower level. With LSI, both indexes performed at equal levels, and it is difficult to say what part of the improvement for the morph index was due to projecting allomorphs to the same dimension and which was due to projecting all related terms. Since the corpus was small, LSI is expected to perform well. However, the queries were long TREC-like sentences and contained many of the words in different inflected forms. This fact makes the best possible improvements smaller.

Query expansion (Section 2.3.5) can also help with the problem of having inconsistent stem morphs. If different inflected forms produce different stem morphs, by adding the alternative stem morphs to the query, recall can be increased. With pseudo relevance feedback, the end result is similar to using LSI, in the sense that co-occurrence of different morphs is the criterion for determining semantic relatedness. In addition to the alternative morphs, the query will be expanded with other terms that have meanings related to the original query terms.

Query expansion for morph-based indexing was tested in Publication II. A parallel blind relevance feedback process was used on a collection of newspaper articles to select the expansion terms. The terms were ranked using Equation 2.15 and a number of best ranked terms were used to expand the queries. It indeed turned out that the expansion terms were often alternative forms of the query stem morphs from different inflected forms. The improvements were much larger than with using LSI: the MAP was rose from 77.7% to 91.8% for the morph index and from 75.7% to 86.6% for the base form index. The additional information provided by the parallel corpus may explain why query expansion works so much better than LSI.

The over- and undersegmentation issue can also be alleviated by determining alternative morph segmentations directly, and using them to expand the query. Since Morfessor uses Viterbi search to find the morpheme segmentation of the query word that is most likely according to the morph frequencies, it is possible to find the n most likely segmentations as well. Adding these alternative segmentations to the query increases the probability that one of them matches the morphs in the recognizer transcripts. In Publication VI, the use of alternative morph segmentation was studied

and the performance of the morph and combined indexes was improved by using 2-best segmentations of query words. At best, for the morph index using short queries, the relative improvement of MGAP was 4.69%.

5.2.4 Selection of Segmentation Units

In Publication VI of this thesis, a TextTiling-based approach for story segmentation of podcasts of Finnish broadcast material was adopted. The method was applied on the ASR transcription using windows of fixed length in time. The level of segmentation can be controlled by varying the parameter α (Section 2.4.1). A text retrieval corpus was used for initial testing by removing all document and sentence structure and placing the documents in random order. The parameter was optimized with respect to resulting retrieval performance, measured by MGAP. It was found that it is best to oversegment the corpus, at optimal level, there were about 50% more segment boundaries than documents in the corpus.

Similarly to retrieval, in segmentation there is a choice of units between morphs, base forms and the combination of the two. For the speech corpus, there were no ground truth story boundaries for training or evaluation. The performance was again measured in terms of resulting MGAP. However, since the indexing units are an other factor that affects the retrieval performance as well, the performance was cross-tested by varying both indexing and segmentation units. The results are summarized in Table 5.4. The selection of segmentation units has a smaller effect than the selection of indexing units. There were statistically significant differences only in one case: when using morph index, the base form segmentation performs significantly worse than morph segmentation. However, on the text corpus, base form segmentation results in better retrieval performance (see Publication VI). This can be explained by the effect on recognition errors. On error free text, the morphological analyzer can find reliable segmentation units, but on the ASR transcripts, the morphs work equally good or better.

It is to be noted that segmentation algorithms behave differently on different data types and some methods may be affected more by recognition errors (Malioutov and Barzilay, 2006). Thus, the tests on the text corpus may not be indicative on how the algorithm performs on a speech corpus. In Publication VI, TextTiling was found to work better in terms of MGAP than segmentation using sliding windows of fixed length in time. On different corpora, sliding windows have been found to give better re-

Table 5.4. Results (MGAP) for cross testing the segmentation and indexing units for the Podcast corpus.

	segmentation		
index	morph	base form	combined
morph	43.8	40.9	42.1
base form	44.7	43.8	45.4
combined	47.0	45.7	46.2

trieval performance (Eskevich et al., 2012). Further analysis on this topic is needed, but differences in the corpora used, e.g. in the granularity of the assigned replay points, are likely explanations of the differences of the results.

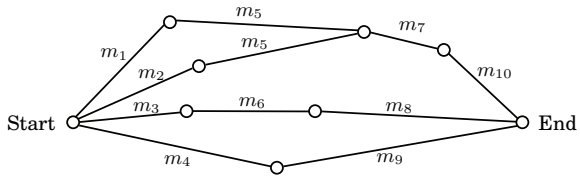
6. Lattices and Confusion Networks

Traditionally, speech recognition aims to minimize the errors in the so-called *1-best* transcription, the most likely hypothesis of the words spoken. However, in speech retrieval, the less likely hypotheses are also a useful source of information. When there is an error in the 1-best transcription, it is possible that the correct word is among the candidates considered by the recognizer. Including these alternative results to the index should improve recall, but the terms will have to be weighted carefully in order not to decrease precision too much.

The alternative candidates that the ASR considers can be extracted in the form of a lattice. When using lattices for speech recognition, one consideration is the level of pruning in the decoder, since it has an effect on the size of the resulting lattices. With a lot of pruning, the resulting lattices are small and easy to process, and there are less spurious results, but some of the correct results may also be pruned. With less pruning, the correct word has a bigger chance of appearing, but there are also more incorrect results, and large lattices are harder to process. The effect of pruning is studied in Publication IV of this thesis.

Lattices tend to be complex structures. The same instance of a word may be represented by multiple arcs that have slightly different time alignments and contexts. An approximation of the lattice that has a simpler form is the *confusion network* (CN) (Mangu et al., 2000). In a confusion network, words that compete around the same time period are clustered together to form a *confusion set*. The words in a confusion set are mutually exclusive and are associated with their posterior probability. The confusion network consists of a non-overlapping series of confusion sets. Sentence hypotheses are obtained by joining word hypotheses selected from each confusion set. The 1-best hypothesis is the one where the word with highest probability is picked from each set. While originally designed

a)



b) Paths:

$$m_1m_5m_7m_{10},\ m_2m_5m_7m_{10},\ m_3m_6m_8,\ m_4m_9$$

c) Confusion network:

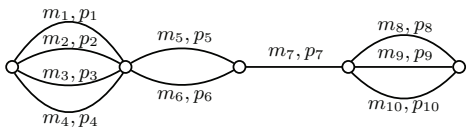


Figure 6.1. a) An ASR lattice. b) All the paths in the lattice. c) The corresponding confusion network. Every morph m_i is associated with its posterior probability p_i.

to minimize the expected WER for ASR, CN also provides a convenient representation of alternative recognition candidates for speech retrieval. Figure 6.1 shows an example of an ASR lattice and the corresponding CN.

The algorithm to generate a confusion network from a lattice has these approximate steps (Mangu et al., 2000):

1. Pruning: The posterior probability for each arc in the lattice is calculated and arcs with very low posterior probability are removed.

2. Intra-word clustering: Arcs that correspond to the same word instance are merged and their probabilities summed.

3. Inter-word clustering: Words that compete around the same time interval are grouped to form confusion sets.

Pruning is needed in order to remove constraining paths from the lattice that would prevent proper alignment of arcs.

In Publications IV, V, and VI of this thesis, using confusion networks is shown to improve Finnish speech retrieval performance. Instead of words, the confusion networks consist of sets of competing morphs that are added to the index weighted based on their posterior probability or rank.

6.1 Using Lattices for Speech Retrieval

Different approaches have been proposed to improve speech retrieval performance by using alternative recognition candidates contained in lattices. In the earliest approaches, a phone recognizer is used to create phone lattices, in which lattice arcs are labeled with phones and associated with their likelihoods (James and Young, 1994; James, 1995; Foote et al., 1997). Keywords are searched from the phone lattices by finding the paths through the lattice that best match the keyword's phonetic representation in a fuzzy comparison. Combining phone lattice-based methods and 1-best word transcripts has been found to perform better than either alone (Jones et al., 1996).

Matching query words in phone lattices requires processing the lattices for the entire collection for each query and is computationally expensive for large collections. For efficient search of speech, an index is first built by counting the frequencies for the terms that appear in the transcriptions. Including alternative recognition candidates requires estimating *expected term frequencies* (ETFs) (also known as expected counts) from the speech recognition lattices either beforehand for all possible terms or for each query word at search time. Saraçlar and Sproat (2004) index phone and word lattices by storing each individual arc, estimate ETFs for query words based on their posterior probability, and return instances where the ETF is above some threshold. However, when using phones, the index is still inefficient, because only the first phone of the query word is located from the index and the subsequent phones are matched by traversing the lattice. Allauzen et al. (2004) index the ETFs of phone strings instead of individual arcs, and estimate the ETF of a query as the minimum ETF of all of its substrings. Yu et al. (2005) extend this approach by estimating the ETFs using m-gram phoneme language models estimated on lattices of segments of audio. A two-stage approach is used: the ETFs are used to select a subset of candidate lattices, and a detailed lattice search is performed on the selected lattices to determine the exact locations. They also use language models for speech recognition that are based on automatically determined phoneme strings. Olsson (2008) uses word and multigram LMs for recognition, converts the resulting lattices to phone lattices, and estimates and indexes ETFs for each phone n-gram sequence $n \leq 5$. Similarly, ETFs can also be estimated for the terms in word lattices as in (Chia et al., 2010), where ETFs are used in a language model based

retrieval that is found to outperform TFIDF and Okapi BM25 in SDR.

Instead of estimating ETFs directly from lattices, the lattices can first be transformed to a simpler form, which will also lead to reduction of required storage space. Siegler (1999) compared using word lattices and n-best list to predict term presence in reference transcripts, and uses the probability of term occurrence as ETF in retrieval. Rank of locally competing terms in lattices was found to correlate better to term presence than the posterior probability of the terms. However, using an ETF based on the fraction of n-best lists that contain the term resulted in overall best performance. Word confusion networks are used for speech retrieval in (Mamou et al., 2006). The posterior probability of the term in the CN is used as the ETF, but for best performance the ETFs are weighted so that more weight is given to terms that have a high rank in the confusion set. Hori et al. (2007) combine word and phone confusion networks. A composition operation for weighted finite-state transducers (WFSTs) is used to align the two networks, and automata intersection is used to match the automaton representation of the query to the confusion networks.

Position specific posterior lattice (PSPL) (Chelba and Acero, 2005; Chelba et al., 2007) is a structure similar to CN, but unlike CNs, it retains the positions of words and can therefore use proximity information for retrieval. For each arc in the lattice, only the position in the lattice (the number of words since the beginning) and the posterior probability for the word in the position are retained. A variation of the standard forward-backward algorithm is used to construct the PSPL (Chelba and Acero, 2005). It is guaranteed that every n-gram present in the lattice is also present in the PSPL. Comparisons of CNs and PSPLs for SDR are performed in (Pan et al., 2007; Pan and Lee, 2010). PSPLs were found to yield better MAP whereas CNs required less storage space.

Using word lattices for retrieval does not help with the OOV-problem, since the lattice will only contain words that are in the vocabulary. Methods based on phone-lattices suffer from the computationally expensive matching of query words to the lattice. In this work, morph lattices, transformed to morph CNs, are used for retrieval. Morph CNs provide a compact representation of alternative recognition candidates, while making possible to match OOV queries by matching the component morphs of the query terms to the morphs in the CN. Matching morphs instead of phones or words also allows taking into account the inflectional properties of the language.

Subwords larger than phonemes are also used in (Mertens and Schneider, 2009), where a fuzzy match between query terms and syllable lattices is used to improve performance of German STD. Parlak and Saraclar (2008) used morphs for Turkish STD by applying the ETF estimation from (Saraçlar and Sproat, 2004) to morph and word lattices. Best performance was achieved when using the morph and word-based methods in cascade: if the word lattices returned no results, the morph lattices were searched. Confusion networks were also tested, and they yielded equal performance to full lattices, but required less storage space. Pan and Lee (2010) recognize speech into standard word lattices, but transform them into subword CNs and PSPLs. Unlike lattices, subword CNs and PSPLs allow combining subword strings into strings that are not substrings of any original IV word. MAP was improved with subword-based indexing for both OOV and IV queries.

6.2 Indexing and Ranking Confusion Networks

In this work, morph and word confusion networks are used for speech retrieval in Finnish (Publications IV, V, and VI). For each spoken document D_i, a lattice is produced with the ASR and then transformed to a CN. Two pieces of information are extracted for each term t at cluster c in the CN: its posterior probability $P(t|c, D_i)$ and its rank in the confusion set $rank(t|c, D_i)$. Two methods for estimating term frequencies are compared. The first method is called *confidence level* or *CL-method*, and it uses the sum of posterior probabilities of all occurrences of the term in the CN:

$$TF(t, D_i) = \sum_{j=1}^{|occ(t,D_i)|} P(t|c_j, D_i), \tag{6.1}$$

where $occ(t, D_i) = (c_1, c_2, \ldots, c_s)$ is the list of all occurrences of t. Thus, the more confident the recognizer is for the term occurring, the more weight the term is given. The formula is the same as used by Mamou et al. (2006), with the exception of a boosting vector that they used to assign more weight to high ranked terms, but which is omitted here.

In the second method, called *rank method*, the term frequency is estimated as the sum of reciprocal ranks of the term in each location:

$$TF(t, D_i) = \sum_{j=1}^{|occ(t,D_i)|} \frac{1}{rank(t|c_j, D_i)}. \tag{6.2}$$

Thus, the term deemed most probable is given the full weight of one, and

the subsequent terms in the confusion set are given less and less weight as their rank increases. In addition to experimental evidence discussed in the following sections, the formula can be motivated by the observation in (Siegler, 1999), where the ranks of locally competing terms in a lattice were found to better correlate with term presence than their posterior probability.

For each term, the inverse document frequency needs also to be estimated. The TFIDF estimates are then used in a vector space model to rank documents using cosine similarity. In Publications IV, and V, IDF based on the number of occurrences of the term in the CN was used. A binary indicator o for term occurrence is estimated by:

$$o(t, D_i) = \begin{cases} 1, & \text{if } TF(t, D_i) > 0 \\ 0, & \text{otherwise} \end{cases} \tag{6.3}$$

The inverse document frequency for a term t is

$$IDF(t) = \log \frac{N}{\sum_i o(t, D_i)}, \tag{6.4}$$

where N is the number of documents in the collection. However, in Publication VI the IDF was changed to:

$$IDF(t) = \log \frac{O}{O_t}, \tag{6.5}$$

where O_t is the sum of estimated TFs for the term t over the entire collection: $O_t = \sum_i TF(t, D_i)$ and $O = \sum_t O_t$. This IDF estimation is the same as used in (Mamou et al., 2006) and was found to perform better on the larger Podcast-corpus.

6.3 Confusion Networks for Morph-based Retrieval

A summary of results for using confusion networks for Finnish speech retrieval is presented in Table 6.1. Expanding the 1-best transcriptions with alternative results from CNs requires weighting the candidates in order not to degrade precision. On both corpora, and for all indexing types, weighting based on the reciprocal rank of the competing terms produces best results. However, the most appropriate weighting is likely dependent on the corpus, the language, and the selection of indexing units.

Two different types of base form CNs have been tested. In Publication V, the lattices were produced with an ASR using traditional word language models. In Publication VI, the lattices were produced with a morph

Table 6.1. Retrieval results comparing 1-best and confusion network indexing using confidence level (CL) or rank based weighting. Morph (m.), base form (b.), combined (c.), and interleaved (i.) indexes were tested (see Section 5.2.2). Results on the Tampere-corpus (Tam.) and the text corpus are mean average precisions (MAP) and results on the Podcast-corpus (Pod.) are mean generalized average precisions (MGAP). The corpora have been tested on normal (N) and high Q-OOV scenarios (H). The results on the latter scenario on the Podcast corpus are previously unpublished. Base form results marked with [1] are produced using a word LM and a word CN. Other base form results are produced using a morph LM, and the resulting morph lattice is transformed into word lattice and further into a word CN.

	Corp.	Tam. (N)	Tam. (N)	Tam. (H)	Pod. (N)		Pod. (H)	
	Ref.	(PIV)	(PV)	(PV)	(PVI)		-	
	Query	long	long	long	long	short	long	short
m.	1-best	76.8	84.4	64.2	43.8	31.8	34.2	19.0
	CL	82.3	-	-	43.7	32.7	36.1	20.9
	rank	85.2	86.9	70.6	46.0	34.6	41.3	26.9
b.	1-best	-	77.9[1]	48.0[1]	43.8	34.7	36.9	21.3
	CL	-	-	-	40.3	32.2	37.8	22.7
	rank	-	80.0[1]	49.8[1]	44.3	35.4	38.5	23.5
c.	1-best	-	-	-	46.2	37.9	40.3	26.1
	CL	-	-	-	45.2	36.1	41.4	26.5
	rank	-	-	-	46.1	37.9	43.8	29.0
i.	1-best	-	-	-	46.8	36.1	38.9	22.6
	CL	-	-	-	46.2	36.1	40.1	23.4
	rank	-	-	-	49.2	37.3	42.4	27.4

language model, and were then transformed into a word lattice. In both cases, the resulting word lattices were transformed into CNs and all the words in the CNs were lemmatized using a morphological analyzer. In both cases, the improvements when using base form CNs are small compared to when using morphs as indexing units. Partly this is due to the fact that, in the morph CNs, different morph segmentations of the same word may appear in the CN. The alternative segmentations increase the likelihood that the query morph matches a morph in the CN. For the word CNs, the alternative segmentations are merged to a single word.

Tampere corpus was used both in Publication IV and V, but changes in recognizer setup (especially acoustic models and pruning parameters) resulted in different recognition results. The recognition error rate and the resulting 1-best retrieval performance were worse in the earlier Publication IV, but relative improvements when using CNs were bigger.

The improvements of using CNs are especially large when the Q-OOV-rate is high. Both morph and base form CN approaches have bigger level of increase in performance than in the normal scenario, but again for the morph index, the improvements are bigger. This shows that even for previously unseen words, the morph recognizer is able to produce morphs that match the query morphs, if not as the most likely candidate, somewhere among the candidate morphs in the CN.

Not surprisingly, the improvements are also higher for short queries than for long queries. This holds for both morph and base form indexes. For long queries, even if a query term is not in the 1-best transcription, there are other query terms that can compensate. For short queries, there are less other terms, and finding the query term in the CN has a bigger significance. Particularly impressive is the 41.6% relative increase in MGAP from 19.0 to 26.9 for the morph index when using short queries in the high Q-OOV scenario.

6.3.1 Effect of Term Weighting

In the Publications of this thesis, speech retrieval experiments were performed using simple cosine similarity ranking with raw term frequencies in the TF component. In initial testing using the small Tampere corpus, raw TF was found to work better than e.g. $log(TF)$ or $(1 + log(TF))$ and therefore chosen as the TF method for subsequent experiments. However, it is possible that a more sophisticated model would yield higher baseline results, and that the simple retrieval model could explain also the im-

provements obtained by using confusion networks. Therefore, some of the experiments were repeated using the more modern Okapi BM25 ranking function (Equation 2.10) with parameter values $b = 0.75$ and $k_1 = 1.2$. The results are listed in Table 6.2. Also reported are the previously unpublished results for different IDFs that resulted in changing the IDF method for the Podcast corpus.

For the Podcast corpus, it can be seen that the BM25 weighting does improve the 1-best baseline results. However, confusion networks still offer improvements in MGAP, from 45.5% to 47.4% with long queries and from 33.0% to 34.2% with short queries (morph index, the best 1-best result is used as a baseline). The former improvement is statistically significant. Using the IDF in Equation 6.5 gives better results than the BM25 IDF (Equation 2.11) both for the CN and for the 1-best results, but for the 1-best results the difference is very small. Using base forms, raw TFs give better results for long queries, but for short queries the BM25 scoring is better. However, in both cases, the differences are small.

For the Tampere corpus, the results are somewhat different. Using the IDF in Equation 6.4 is significantly better than either the one in Equation 6.5 or BM25 scoring for the morph index, but for the base form index BM25 is slightly better.

It can be concluded that both the corpus and the selection of index terms have an effect on what is the most appropriate weighting of index terms. The very small size and artificial nature of the Tampere corpus most likely explains some of the differences in results. In addition, the corpus contains only planned speech in noise free conditions, making it relatively easy as a recognition task. As a result, there are fewer alternative candidates in the confusion networks.

The full effect of the ranking function, its parameters, and the IDF function is a question that still warrants further research.

Table 6.2. Retrieval results (MAP/MGAP) comparing different TF and IDF weightings. Results for the Tampere corpus (Tam.) are comparable to those of Publication V and results for the Podcast corpus (Pod.) to Publication VI. Cosine ranking (Equation 2.3) with raw term frequencies, and Okapi BM25 scoring (Equation 2.10) were tested. The two CN IDF weightings were tested: Equation 6.4 and Equation 6.5. With BM25 scoring, the commonly used IDF for Okapi BM25 in Equation 2.11 was used. The BM25 IDF was floored to zero.

Morph

TF	IDF	index	Tam.	Pod. (long)	Pod. (short)
raw	Eq. 6.4	1-best	84.4	43.4	31.6
raw	Eq. 6.4	CL	86.9	40.1	32.7
raw	Eq. 6.4	rank	86.9	43.1	30.9
raw	Eq. 6.5	1-best	75.4	43.8	31.8
raw	Eq. 6.5	CL	76.3	43.7	32.7
raw	Eq. 6.5	rank	77.9	46.0	34.6
BM25	Eq. 2.11	1-best	81.7	45.0	33.0
BM25	Eq. 2.11	CL	85.5	44.6	31.2
BM25	Eq. 2.11	rank	84.5	46.5	33.0
BM25	Eq. 6.5	1-best	76.3	45.5	32.7
BM25	Eq. 6.5	CL	76.7	46.4	33.4
BM25	Eq. 6.5	rank	78.5	47.4	34.2

Base form

TF	IDF	index	Tam.	Pod. (long)	Pod. (short)
raw	Eq. 6.4	1-best	77.9	41.3	31.7
raw	Eq. 6.4	CL	78.7	34.1	23.9
raw	Eq. 6.4	rank	80.0	38.1	27.0
raw	Eq. 6.5	1-best	72.1	43.8	34.7
raw	Eq. 6.5	CL	73.6	40.3	32.2
raw	Eq. 6.5	rank	73.8	44.3	35.4
BM25	Eq. 2.11	1-best	78.8	43.4	35.1
BM25	Eq. 2.11	CL	80.3	39.0	29.2
BM25	Eq. 2.11	rank	80.3	41.9	32.0
BM25	Eq. 6.5	1-best	75.7	42.9	36.3
BM25	Eq. 6.5	CL	76.4	41.7	35.2
BM25	Eq. 6.5	rank	77.0	44.2	37.4

7. Evaluation Metrics for Unsupervised Morphological Analysis

Speech retrieval and other natural language processing applications need morphological analysis methods for processing inflected word forms. As seen in preceding chapters, using inflected word forms as language modeling or information retrieval units does not work well for morphologically rich languages such as Finnish. Unsupervised morphological analysis methods such as Morfessor can be used for languages for which rule-based analyzers do not exist, and they also provide improvements for languages that do have them, for example by not limiting the lexicon of the analyzer. While usefulness of a morphological analysis method is ultimately decided by the application where it is used, application evaluations tend to be time-consuming and can not be used during method development. Another option is to compare the analyses provided by the method with manually created reference analyses. In full morphological analysis, all morphemes in each word should be listed. For example, the linguistic reference of the word "giving" can be given as "give_V +PCP1". However, for unsupervised methods, the method has no way of knowing which sort of morpheme labels are used in the references, and should not be expected to produce labels that are the same as the ones created by linguists. Thus, the evaluation of morphological analysis methods is a challenging research problem in itself.

One criterion for an evaluation metric is that it should correlate with the performance of target applications that use morphological analysis. The Morpho Challenge (Kurimo et al., 2006a, 2008, 2009a,b, 2010a,b) is a series of competitions that aims to evaluate and design unsupervised (or semi-supervised) and language independent methods for learning morphology from large collection of text data. Participants have provided morphological analyses for given word lists, which have then been evaluated in linguistic and application tasks. Speech recognition, information re-

trieval and machine translation have been tested over the years. In Publication VII, the results over the years are collected and used in a large meta-analysis covering a large number of analysis methods and different languages and tasks. The results are used to empirically compare different linguistic evaluation methods of morphological analysis. Particularly, the correlation of the metrics to application performance is examined.

7.1 Evaluation by Linguistic Comparison

In direct evaluations of morphological analysis methods, the proposed word analyses are examined manually by experts, or more typically, compared to linguistic "gold standard" references by automatic means. The simplest evaluations only consider the boundaries in segmented word forms (Hafer and Weiss, 1974; Kurimo et al., 2006a), but in general, evaluations of full morphological analysis is desired (Kurimo et al., 2008; Spiegler and Monson, 2010). In both cases, the evaluation involves calculating the precision (p, the proportion of predicted morphemes/boundaries that were also in the reference) and the recall (r, the proportion of the reference morphemes/boundaries that were predicted) of the method. Methods that tend to oversegment will have high recall but low precision, and methods that tend to undersegment will have high precision but low recall. A commonly used measure that captures both precision and recall is the F_β*-score*:

$$F_\beta = \frac{(1+\beta^2)pr}{\beta^2 p + r}, \tag{7.1}$$

where $\beta > 1$ gives more weight to recall and $\beta < 1$ gives more weight to precision. When $\beta = 1$, the measure is called simply the *F-score*.

Since the predicted morphemes can be arbitrary, simply examining the sets of morphemes is not possible. Instead, *co-occurrence analysis* is used, where it is examined how morphemes are shared between words. The approach can also be called *isomorphic analysis* (Spiegler and Monson, 2010) since the problem is related to solving graph *isomorphism*: the analyses for a set of words can be represented as *bipartite graph* $G = (M, W; E)$, which has two disjoint sets of vertices, morphemes $M = \{m_1, \ldots, m_n\}$ and words $W = \{w_1, \ldots, w_m\}$, as well as edges $e(m_i, w_j)$ that connect vertices in M to vertices in W. If two graphs are isomorphic, that is, there exists a bijection between their vertices, the corresponding sets of analyses are equivalent. *Word graph* is another representation of the analyses,

in which the morpheme vertices are disregarded and each pair of edges $e(m_i, w_j), e(m_i, w_k)$ is replaced by edge $e(w_j, w_k)$ so that two words are connected if they share a morpheme.

Comparing the word graphs of the reference analyses and the predicted analyses, recall can be calculated as the proportion of edges that are in the reference graph but not in the predicted graph and precision as the proportion of edges that are in the predicted graph but not in the reference graph. However, comparing large graphs as a whole can be computationally inefficient, and some form of approximation may have to be applied.

7.1.1 MC-metric

For Morpho Challenge 2007 (Kurimo et al., 2008), an evaluation metric (*MC-metric*) for full morphological analysis was developed, and in 2009 it was slightly revised (Kurimo et al., 2010b). In the evaluation, a set of random word pairs that have at least one morpheme in common is sampled: first a number of *focus words* are sampled, and then for each predicted morpheme from each focus word, another word that has the same morpheme is sampled. This corresponds to sampling edges from the word graph. For each focus word, the precision is the proportion of its word pairs that also have a common morpheme in the reference analyses. Recall is calculated similarly by sampling focus words and their pairs from the reference. Overall precision and recall scores are the average over focus words. Alternative analyses for possibly ambiguous word forms are allowed. If there are multiple analyses for a word, the precision is the average of precisions of different alternatives. If there are multiple references, the precision is the maximum of precisions. The same holds for recall, but in a mirror fashion.

The MC-metric is efficient to calculate, but it has a few limitations. Since the sampled word pairs are dependent on the predicted analyses, different algorithms will have different evaluation sets. Also, not all information in the known analyses are used, which will lead to inefficient estimates if the reference set is small. The MC-metric is also susceptible to different types of *gaming* as demonstrated by Spiegler and Monson (2010). In *ambiguity hijacking*, unreasonable amount of alternative segmentations are given as if there was great ambiguity in all words. The MC-metric will give a high F-score to these systems. In *shared morpheme padding*, the same bogus morpheme is added to analysis of every word. This will greatly increase recall scores. Precision will drop, but the result-

ing F-score can still be unreasonably high.

7.1.2 EMMA

MC-metric can be called a *soft* isomorphic measure, as it does not map morphemes in the reference and predicted analyses to each other. Spiegler and Monson (2010) proposed a *hard* isomorphic measure called *EMMA* (evaluation metric for morphological analysis), which seeks one-to-one mapping between the predicted and reference morphemes. After the optimal mapping is found, precision and recall can be directly calculated based on the number of shared morphemes.

The mapping is found using bipartite graphs, where one set of vertices correspond to the reference morphemes and another set to the predicted ones. If there is a word that has in its reference analysis the morpheme m_i and in the predicted analysis the morpheme m_j, an edge $e(m_i, m_j)$ exists between the morphemes. The weight c_{ij} of the edge is the number of such words. The more weight the edge has, the bigger the resulting precision and recall is, if the edge is selected in the final mapping. Thus, the task is to select the edges in the bipartite graph, so that a one-to-one mapping is realized and that the sum of the weights of the selected edges is maximized. The maximization can be achieved using integral linear programming techniques by defining a binary matrix $\mathbf{B}$ where $b_{ij} = 1$ indicates that there is an edge from m_i to m_j in the optimal solution. The problem can be then defined as:

$$\underset{\mathbf{B}}{\operatorname{argmax}} \sum_{i,j} (c_{ij} \times b_{ij}) \quad \text{s.t.} \quad \sum_i b_{ij} \leq 1, \sum_j b_{ij} \leq 1, b_{ij} \in \{0, 1\}. \tag{7.2}$$

Alternative analyses are allowed in EMMA by normalizing the weight of morpheme pairs by the number of possible pairings between proposed and reference analysis alternatives. After finding the one-to-one morpheme mapping, bipartite graphs are used again, this time to match full proposed and reference analysis alternatives so that the number of correctly predicted morphemes is maximized across all alternatives.

The one-to-one matching of EMMA means that the metric accounts for allomorphy and syncretism by penalizing analyses that do not map allomorphs to the same morpheme or differentiate between syncretic morphemes that have identical forms but different meanings. The provided mappings allow to examine the performance of different algorithms qualitatively. Experiments also show that EMMA is not vulnerable to gaming like the MC-metric (Spiegler and Monson, 2010).

However, the computational cost of solving the bipartite graphs grows rapidly when the size of the linguistic reference is increased. In Publication VII of this thesis, a modified version of EMMA (*EMMA-2*) was proposed that reduces the computational complexity by replacing the one-to-one mapping by two many-to-one assignments. The idea is that if two allomorphs are not joined, it affects recall but not precision, and if two syncretic morphemes are not distinguished, it affects precision but not recall. Thus, when calculating precision, a many-to-one mapping is used where several predicted morphemes may be assigned to the same reference morpheme. Similarly, when calculating recall, a one-to-many mapping is used. In EMMA-2, the matching equations are

$$\mathbf{B}_p = \underset{\mathbf{B}}{\operatorname{argmax}} \sum_{i,j} (c_{ij} \times b_{ij}) \quad \text{s.t.} \quad \sum_j b_{ij} \leq 1, b_{ij} \in \{0,1\}, \tag{7.3}$$

for precision and

$$\mathbf{B}_r = \underset{\mathbf{B}}{\operatorname{argmax}} \sum_{i,j} (c_{ij} \times b_{ij}) \quad \text{s.t.} \quad \sum_i b_{ij} \leq 1, b_{ij} \in \{0,1\}, \tag{7.4}$$

for recall. In both cases, the best match for each morpheme can be selected independently from others, reducing the computational complexity significantly. However, the possibility for gaming is increased.

7.1.3 CoMMA

Another new metric (*CoMMA*, co-occurrence-based metric for morphological analysis) was proposed in Publication VII. It is based on the MC-metric and aims to deal with some of the limitations of the MC-metric. Namely, that MC-metric uses different word pairs for different algorithms, and that not all references are used in the calculation. Co-occurrence between morphemes in the reference answers are counted in a matrix $\mathbf{A}$, and in the suggested analyses in a matrix $\mathbf{S}$. First, the case where there are no alternative analyses for words is considered. The matrix elements a_{ij} and s_{ij} are the number of morphemes that are shared between words i and j in the reference and in the suggested analyses, respectively. The difference between the matrixes tells us about the quality of the analysis. Let the numbers of words that share at least one morpheme with the word i be $n_i = |\{j : s_{ij} > 0\}|$ and $m_i = |\{j : a_{ij} > 0\}|$. The numbers of words that have at least one common morpheme with any word is $v_s = |\{i : n_i > 0\}|$

and $v_a = |\{i : m_i > 0\}|$. The precision and recall can be given as:

$$p = \frac{1}{v_s} \sum_{i:n_i>0} \frac{1}{n_i} \sum_{j:s_{ij}>0} \frac{\min(s_{ij}, a_{ij})}{s_{ij}}; \quad (7.5)$$

$$r = \frac{1}{v_a} \sum_{i:m_i>0} \frac{1}{m_i} \sum_{j:a_{ij}>0} \frac{\min(a_{ij}, s_{ij})}{a_{ij}}. \quad (7.6)$$

Two methods were tested for extending the metric to the case where several alternative analyses are allowed. In the first (*CoMMA-B*) the maximal co-occurrence count is taken for each word. Adding more alternatives will generally increase recall but degrade precision. In the second approach (*CoMMA-S*), wrong number of alternatives is directly penalized. The alternatives are added to the rows of the matrixes **S** and **A**. Calculating precision and recall now involve solving an assignment between predicted and reference alternatives that optimizes the F-score. See Publication VII for details.

It was also tested whether isolated words that do not share morphemes with any other word should be excluded from evaluation. The exclusion is done by setting the diagonals of matrixes A and S to zero. "0" in the algorithm name is used to mark exclusion and "1" inclusion. The tested alternatives were thus: CoMMA-B0, CoMMA-B1, CoMMA-S0, and CoMMA-S1.

7.2 Application Evaluations

In indirect evaluations, the morphological analysis methods are compared on the basis of how well they perform in a real NLP task. In Morpho Challenge, multiple tasks in multiple languages have been used to evaluate morphological analysis algorithms.

7.2.1 Information Retrieval

The information retrieval task is similar to the morph-based approach for speech retrieval used in this thesis. All the words in the corpus and in the queries are replaced by their suggested analyses, and thus queries are matched to documents based on morphemes instead of words. A number of reference methods were also tested: words without any analysis, base forms by a rule-based analyzer, and stemming. In information retrieval, finding stem morphemes is most important, since all the documents that contain the query word in any inflected word should be returned. If the evaluated algorithm uses special markings for affixes they would be easy

to exclude. However, another approach was used, where an automatic stop list is constructed by excluding morphemes that have more occurrences than a certain threshold. The generated stop lists contained mostly affix morphemes. This way, all algorithms can be treated equally. Using Okapi BM25 for ranking, the performance without a stop list was severely degraded. By looking at Equation 2.11, it is apparent why: if a morpheme is present in more than half of the documents, the IDF for the morpheme is negative and almost any other document will be ranked higher than a document with the morpheme. Another option would be to limit the IDF to always be higher than zero or a constant near zero.

The experiments were performed on Finnish, English and German corpora. In all languages, the best MAP was achieved by one of the language-specific reference algorithms. For German and Finnish, the two-level morphological analyzer gave the best results, and for English the traditional Porter stemmer (Kurimo et al., 2010a). The performance of the best unsupervised methods were very close to the reference methods, and a number of algorithms achieved a level of performance that was not significantly worse than the reference methods. Achieving statistically significant differences is in fact difficult with the limited number of queries, and the results should be interpreted with care.

Some conclusions can be drawn, however. The Morfessor Baseline algorithm performed reasonably well in all languages. Also, combining results from two different algorithms is a good strategy for maximal IR performance. The ParaMor method by Monson et al. (2008) uses an unsupervised model for building linguistically motivated paradigms. Combined with the results of Morfessor CatMAP, very good IR results are achieved. An extension of Morfessor Baseline, *Allomorfessor* (Kohonen et al., 2009), models allomorphy by allowing strings mutations, and improves the IR results for Finnish over Morfessor Baseline. Perhaps disappointingly for developers of morphological analysis, using simple letter n-grams (McNamee and Mayfield, 2007) gives excellent IR results, even for the morphologically very complex Finnish language.

7.2.2 Speech Recognition

In speech recognition, morpheme segmentation is needed to build efficient language models for morphologically rich languages. In (Kurimo et al., 2006a), n-gram language models for Finnish and Turkish were built using corpora segmented with different algorithms and the resulting speech

recognition error rates were compared. It was noted that it is hard to achieve significant differences between methods. The experiments were also very labor intensive, and they were not repeated in later challenges.

7.2.3 Machine Translation

In statistical machine translation (SMT), statistical models are estimated from bilingual text corpora, and used to translate sentences from a source language to a target language. By incorporating morphological analysis in SMT, potentially better translation models can be built, since data sparsity in training data is reduced, for example by taking into account dependencies between inflected word forms (Nießen and Ney, 2004). German to English and Finnish to English translation systems were trained (Kurimo et al., 2010b). Morphological analyses were applied to the words in the source languages but not in the target language. The results show that using morphological analysis does not improve the results over a word-based baseline. However, combining morpheme-based and word-based approaches, the best algorithms perform significantly better than the word-based method alone.

7.3 Metric Correlations

In Publication VII, the correlations of different linguistic evaluation metrics with respect to IR and SMT performance were examined. Morphological analysis results of about 20 different methods or their variants were available. The performance of the analysis methods were evaluated using different linguistic metrics, as well as in IR and SMT application tasks. Further, the evaluation metrics were evaluated in terms of susceptibility to gaming, computational complexity and stability to changes in the evaluation set. Selected results concentrating on the IR aspects are reviewed here.

Figure 7.1 shows the correlation of F-score from different linguistic metrics to MAP in IR using Spearman's rank correlation. EMMA correlates highly in English and Finnish, and moderately in German. The EMMA-2 metric gives very close results. The MC-metric performs poorly in English and Finnish, but surprisingly the correlation is the highest in German. The CoMMA metrics perform well for English, the S1-variant is almost at par with EMMA, but for Finnish their performance is poor and for Ger-

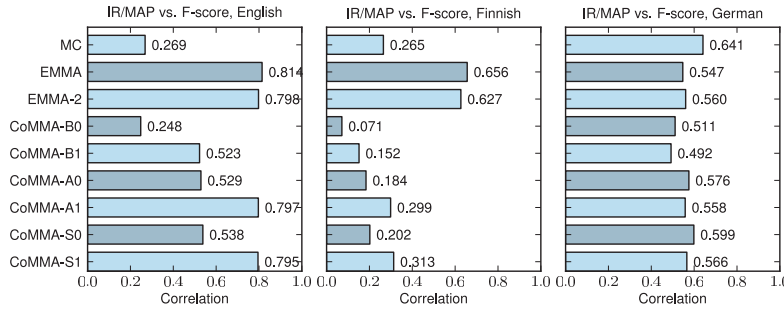


Figure 7.1. Correlations between the F-scores of different linguistic evaluation methods and the scores of IR and SMT evaluations. Figure lifted from Publication VII.

man moderate.

The F-score weighs precision and recall equally. However, it is not certain that in information retrieval precision and recall in morphological analysis have equal importance in the resulting MAP. The tradeoff between precision and recall is related to how aggressively the algorithm segments the words. If the algorithm undersegments, the precision will be high but recall low. In IR, the effect is similar: recall will drop if related word forms are not conflated. The reverse holds for oversegmentation: recall will rise and precision will drop. Figure 7.2 plots how F_β correlates to MAP in IR with different values of β. For the co-occurrence based MC and CoMMA-metrics, precision plays more important role than recall as the maximum values are achieved with $\beta < 1$. For EMMA and EMMA-2, the maximum values are reached when recall is weighted more, but the balanced F-score at $\beta = 0$ is also very good. Overall, with the optimally weighted versions, the performance of MC and CoMMA metrics is brought much closer to EMMA and EMMA-2. The graph for German is slightly strange, since very good correlations are achieved also when only precision is taken into account.

The results show that EMMA has a very good overall performance. It has a high correlation with application performance, it is hard to game, and the one-to-one mapping of morphemes provides opportunity to interpret the results qualitatively. Its weakness is the computational complexity. EMMA-2 solves the computational complexity problem, while maintaining the good correlation and robustness of EMMA. The drawback of EMMA-2 is that it does not provide as clear mapping of morphemes. However, it provides more interpretable view of precision and recall in cases where the number of predicted morphemes is wrong. The CoMMA meth-

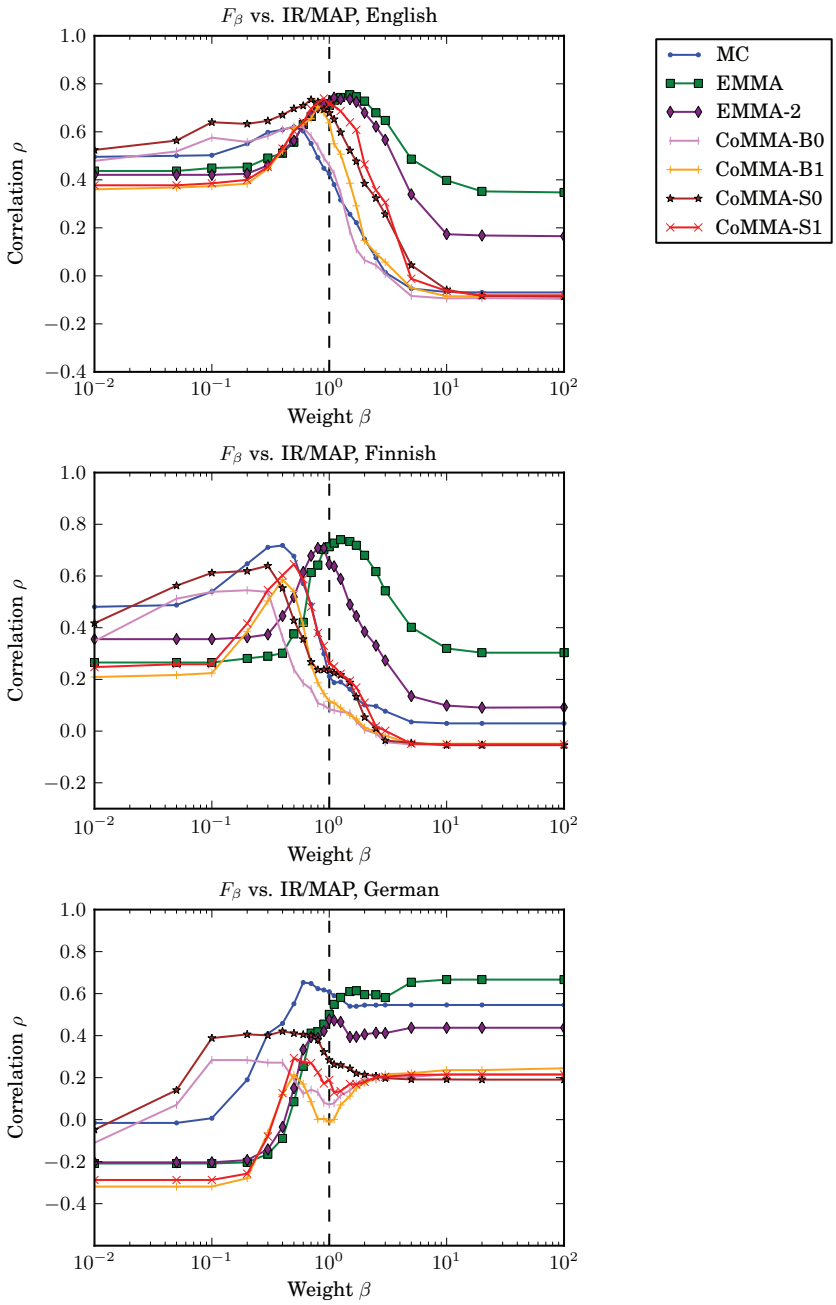


Figure 7.2. Correlations between the results of the application evaluations and weighted F_β-scores as a function of β. Figure lifted from Publication VII.

ods improve over the MC-metric, but lose to the EMMA metrics in terms of correlation to application performance. However, for English the metrics work as well as EMMA.

8. Conclusions

This thesis proposes to use morph-based methods for speech retrieval. Morpheme-like units are discovered statistically from unlabeled text and used for language modeling, story segmentation and retrieval. The speech retrieval experiments are performed on Finnish, a highly agglutinative language. While Finnish is likely to benefit more from the methods proposed in this thesis, other languages, even morphologically simple ones, could benefit from the improved ability to recognize OOV-words. Compared to many other OOV robust retrieval methods, the biggest advantage of the morph-based approach is its simplicity: there is no need for traversing lattices at query time, instead any text retrieval method can be applied to the morph frequency estimates from transcripts or from confusion networks. However, morph-based retrieval on other languages still warrants further research.

The results show that using a morph-based language model greatly improves speech retrieval performance, even when compared to a very large word language model. Further, even on high quality speech transcripts, using morphs as index terms works as well as the traditional method of using base forms provided by a rule-based morphological analyzer. If an analyzer is available, combining the morph and base form based approaches is the recommended method. The results on the scenarios where the Q-OOV-rate is high, are particularly illustrative of the benefits of the combined approach. Morphs and base forms both capture different features that neither of them can capture alone.

In story segmentation, morphs and base forms perform at about equal levels. However, comparing the performance between segmenting error-free text and ASR transcripts, it seems that morphs are somewhat more resilient to recognition errors than base forms.

Extracting alternative recognition candidates from confusion networks

further improves retrieval results. The increase is larger for morph-based indexing. Partly this is because it compensates for a weakness of the morph-based approach: in some cases there are multiple possible morph boundaries for a word, and one of them will appear in the transcription, and the other in the query. Using confusion networks, some of the alternatives can appear in the confusion set. However, in the word-based indexing there are also improvements, especially in the high Q-OOV scenario, showing that some of the candidates in the confusion network are legitimate alternative results, and including them in the index improves retrieval performance. Further, confusion networks provided improvements also when the queries were directly expanded with alternative morph segmentations. Weighting the results based on the reciprocal rank of the term was found more effective than based on the probability. However, it was also discovered that the corpus size has an effect on the proper weighting.

In this thesis, speech recognition, story segmentation, indexing and retrieval parts of a speech retrieval system are considered. There are still points of improvement and research in all those areas. In segmentation, incorporating acoustic cues would likely improve the results, and using alternative lexical methods could also be tested. In speech recognition, the language model tries to capture sequences of morphs, while in retrieval, the interest lies in lemmas that capture the content of the speech. In this work, the same set of morphs are used for both, but the set of morphs that works best for the former, will not necessarily work best for the latter. Using the same set of morphs for both recognition and retrieval is justified, however, by the fact how unseen words are recognized. Some of the morphs in the word can be recognized correctly, but usually not the entire word, thus it is important to segment the query word using the same morphs. There is a tradeoff between accurate recognition of common words, and being able to include also rare and unseen words. For example, shorter morphs would give a better chance of recognizing unseen or rare words, but shorter morphs would also degrade the recognition of common words, which would have a negative effect on retrieval performance as well. Thus, the selection of language modeling units that provides best retrieval results also warrants some further research.

In the morph-based system, because the units are shorter, the dependency between adjacent units in the transcript is larger than in a word-based system. However, the vector space model used in this thesis ignores

the proximity information completely. A possibly better alternative would be using a retrieval method based on the language modeling approach. Estimating the models from confusion networks would allow a natural way of incorporating alternative recognition candidates as well.

For text retrieval, the morph-based approach is viable as well, and while improvements in statistical machine translation are harder to achieve, they are also possible. While the advantages over rule-based methods may not be as clear as in the case of retrieving OOV-words from speech, the results show that Morfessor and other unsupervised methods can provide good performance. Especially less resourced languages, and dialectal or colloquial language material can benefit from unsupervised methods. The morph-based approach has potential in other natural language processing applications as well. In future work, using morphs for part-of-speech tagging, speech synthesis and word sense disambiguation will be tested. In this thesis, methods for evaluation of unsupervised learning of morphology are compared empirically. An evaluation method that correlates well with real application performance, and that is readily computable, will help in development of unsupervised morphological analysis methods. Those, in turn, will help improve performance of applications such as text or speech retrieval.

Bibliography

E. Agirre, G. Di Nunzio, N. Ferro, T. Mandl, and C. Peters. CLEF 2008: Ad hoc track overview. In C. Peters, T. Deselaers, N. Ferro, J. Gonzalo, G. Jones, M. Kurimo, T. Mandl, A. Peñas, and V. Petras, editors, *Evaluating Systems for Multilingual and Multimodal Information Access*, volume 5706 of *Lecture Notes in Computer Science*, pages 15–37. Springer Berlin / Heidelberg, 2009. ISBN 978-3-642-04446-5.

A. Ajanki, M. Billinghurst, T. Jarvenpaa, M. Kandemir, S. Kaski, M. Koskela, M. Kurimo, J. Laaksonen, K. Puolamaki, T. Ruokolainen, and T. Tossavainen. Contextual information access with augmented reality. In *Proceedings of the IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 95–100, Sept. 2010.

M. Akbacak, D. Vergyri, and A. Stolcke. Open-vocabulary spoken term detection using graphone-based hybrid recognition systems. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'08)*, pages 5240–5243, Apr. 2008.

C. Allauzen, M. Mohri, and M. Saraclar. General indexation of weighted automata: application to spoken utterance retrieval. In *Proceedings of the Workshop on Interdisciplinary Approaches to Speech Indexing and Retrieval at HLT-NAACL 2004*, SpeechIR '04, pages 33–40, Stroudsburg, PA, USA, 2004. ACL.

E. Arisoy, H. Dutağaci, and L. M. Arslan. A unified language model for large vocabulary continuous speech recognition of Turkish. *Signal Process.*, 86(10): 2844–2862, Oct. 2006. ISSN 0165-1684.

B. S. Atal. Effectiveness of linear prediction characteristics of the speech wave for automatic speaker identification and verification. *J. Acoust. Soc. Am*, 55(6): 1304–1312, 1974.

A. Atreya and C. Elkan. Latent semantic indexing (LSI) fails for TREC collections. *SIGKDD Explor. Newsl.*, 12(2):5–10, Mar. 2011. ISSN 1931-0145.

D. Beeferman, A. Berger, and J. Lafferty. Statistical models for text segmentation. *Machine learning*, 34(1):177–210, 1999.

M. Bisani and H. Ney. Open vocabulary speech recognition with flat hybrid models. In *Proceedings of the 9th European Conference on Speech Communication and Technology (Interspeech 2005 – Eurospeech)*, pages 725–728, 2005.

D. M. Blei and P. J. Moreno. Topic segmentation with an aspect hidden markov model. In *Proceedings of the 24th annual international ACM SIGIR conference*

on Research and development in information retrieval, pages 343–348, New York, NY, USA, 2001. ACM. ISBN 1-58113-331-6.

R. B. Bradford. An empirical study of required dimensionality for large-scale latent semantic indexing applications. In *Proceeding of the 17th ACM conference on Information and knowledge management*, CIKM '08, pages 153–162, New York, NY, USA, 2008. ACM. ISBN 978-1-59593-991-3.

M. G. Brown, J. T. Foote, G. J. F. Jones, K. Spärck Jones, and S. J. Young. Open-vocabulary speech indexing for voice and video mail retrieval. In *Proceedings of the fourth ACM international conference on Multimedia*, MULTIMEDIA'96, pages 307–316, New York, NY, USA, 1996. ACM. ISBN 0-89791-871-1.

C. Chelba and A. Acero. Position specific posterior lattices for indexing speech. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05, pages 443–450, Stroudsburg, PA, USA, 2005. ACL.

C. Chelba, J. Silva, and A. Acero. Soft indexing of speech content for search in spoken documents. *Comput. Speech Lang.*, 21(3):458–478, 2007.

B. Chen, H. min Wang, and L. shan Lee. Discriminating capabilities of syllable-based features and approaches of utilizing them for voice retrieval of speech information in Mandarin Chinese. *IEEE Transactions on Speech and Audio Processing*, 10(5):303–314, July 2002.

S. F. Chen and J. Goodman. An empirical study of smoothing techniques for language modeling. *Computer Speech & Language*, 13(4):359–393, 1999. ISSN 0885-2308.

T. K. Chia, K. C. Sim, H. Li, and H. T. Ng. Statistical lattice-based spoken document retrieval. *ACM Trans. Inf. Syst.*, 28(1):1–30, 2010. ISSN 1046-8188.

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

F. Y. Y. Choi, P. Wiemer-Hastings, and J. Moore. Latent semantic analysis for text segmentation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processin (EMNLP)*, pages 109–117, 2001.

N. Chomsky and M. Halle. *The sound pattern of English*. Harper & Row, New York, 1968.

G. Choueiter, D. Povey, S. F. Chen, and G. Zweig. Morpheme-based language modeling for Arabic lvcsr. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'06)*, pages 1053–1056, 2006.

M. Creutz. *Language Models for Automatic Speech Recognition*. PhD thesis, Helsinki University of Technology, 2007.

M. Creutz and K. Lagus. Unsupervised discovery of morphemes. In *Proceedings of the ACL-02 workshop on Morphological and phonological learning*, MPL '02, pages 21–30, Stroudsburg, PA, USA, 2002. ACL.

M. Creutz and K. Lagus. Induction of a simple morphology for highly-inflecting languages. In *Proceedings of the 7th Meeting of the ACL Special Interest Group in Computational Phonology: Current Themes in Computational Phonology and Morphology*, SIGMorPhon '04, pages 43–51, Stroudsburg, PA, USA, 2004. ACL.

M. Creutz and K. Lagus. Inducing the morphological lexicon of a natural language from unannotated text. In *Proceedings of the International and Interdisciplinary Conference on Adaptive Knowledge Representation and Reasoning (AKRR'05)*, pages 106–113, 2005a.

M. Creutz and K. Lagus. Unsupervised morpheme segmentation and morphology induction from text corpora using Morfessor 1.0. Technical Report A81, Publications in Computer and Information Science, Helsinki University of Technology, 2005b. `http://www.cis.hut.fi/projects/morpho/`.

M. Creutz and K. Lagus. Unsupervised models for morpheme segmentation and morphology learning. *ACM Trans. Speech Lang. Process.*, 4(1):3:1–3:34, Feb. 2007. ISSN 1550-4875.

M. Creutz, T. Hirsimäki, M. Kurimo, A. Puurula, J. Pylkkönen, V. Siivola, M. Varjokallio, E. Arısoy, M. Saraçlar, and A. Stolcke. Morph-based speech recognition and modeling of out-of-vocabulary words across languages. *ACM Trans. Speech Lang. Process.*, 5(1):1–29, 2007. ISSN 1550-4875.

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

S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407, 1990.

S. Deligne and F. Bimbot. Inference of variable-length acoustic units for continuous speech recognition. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'97)*, pages 1731–1734, Apr. 1997.

S. T. Dumais. Latent semantic indexing (LSI): TREC-3 report. In *Overview of the Third Text REtrieval Conference*, pages 219–230, 1995.

I. Ekman. Suomenkielinen puhehaku (Finnish spoken document retrieval). Master's thesis, University of Tampere, Finland, 2003. (in Finnish).

A. El-Desoky, C. Gollan, D. Rybach, R. Schlüter, and H. Ney. Investigating the use of morphological decomposition and diacritization for improving Arabic LVCSR. In *Proceedings of 10th Annual Conference of the International Speech Communication Association (Interspeech 2009)*, pages 2679–2682, Sept. 2009.

A. El-Desoky Mousa, M. A. B. Shaik, R. Schlüter, and H. Ney. Sub-lexical language models for German LVCSR. In *Proceedings of the IEEE Spoken Language Technology Workshop (SLT)*, pages 171–176, Dec. 2010.

A. El-Desoky Mousa, M. A. B. Shaik, R. Schlüter, and H. Ney. Morpheme based factored language models for German LVCSR. In *Proceedings of the 12th Annual Conference of the International Speech Communication Association (Interspeech 2011)*, pages 1445–1448, Aug. 2011.

M. Eskevich, W. Magdy, and G. J. F. Jones. New metrics for meaningful evaluation of informally structured speech retrieval. In R. Baeza-Yates, A. de Vries, H. Zaragoza, B. Cambazoglu, V. Murdock, R. Lempel, and F. Silvestri, editors, *Advances in Information Retrieval*, volume 7224 of *Lecture Notes in Computer Science*, pages 170–181. Springer Berlin / Heidelberg, 2012. ISBN 978-3-642-28996-5.

J. G. Fiscus, J. Ajot, J. S. Garofolo, and G. Doddingtion. Results of the 2006 spoken term detection evaluation. In *Proceedings of the ACM SIGIR Workshop 'Searching Spontaneous Conversational Speech'*, pages 51–57, July 2007. ISBN 978-90-365-2542-8.

J. Foote, S. Young, G. Jones, and K. Spärck Jones. Unconstrained keyword spotting using phone lattices with application to spoken document retrieval. *Computer Speech & Language*, 11(3):207–224, 1997. ISSN 0885-2308.

J. T. Foote, G. J. F. Jones, K. Spärck Jones, and S. J. Young. Talker-independent keyword spotting for information retrieval. In *Proceedings of the 4th European Conference on Speech Communication and Technology (Eurospeech'95)*, Sept. 1995.

M. Gales. Maximum likelihood linear transformations for HMM-based speech recognition. *Computer Speech & Language*, 12(2):75–98, 1998.

M. Galley, K. McKeown, E. Fosler-Lussier, and H. Jing. Discourse segmentation of multi-party conversation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics (ACL'03)*, pages 562–569, Stroudsburg, PA, USA, 2003. ACL.

J. S. Garofolo, C. G. P. Auzanne, and E. M. Voorhees. The TREC spoken document retrieval track: A success story. In *Content-Based Multimedia Information Access: Recherche d'Informations Assistee par Ordinateur (RIAO)*, pages 1–20, 2000.

M. A. Hafer and S. F. Weiss. Word segmentation by letter successor varieties. *Information Storage and Retrieval*, 10(11-12):371–385, 1974.

M. A. Hearst. TextTiling: segmenting text into multi-paragraph subtopic passages. *Comput. Linguist.*, 23(1):33–64, 1997. ISSN 0891-2017.

I. L. Hetherington. *A characterization of the problem of new, out-of-vocabulary words in continuous-speech recognition and understanding*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, 1995.

T. Hirsimäki. *Advances in Unlimited-Vocabulary Speech Recognition for Morphologically Rich Languages*. PhD thesis, Helsinki University of Technology, 2009.

T. Hirsimäki, M. Creutz, V. Siivola, M. Kurimo, S. Virpioja, and J. Pylkkönen. Unlimited vocabulary speech recognition with morph language models applied to Finnish. *Computer Speech & Language*, 20(4):515–541, 2006.

T. Hirsimäki, J. Pylkkönen, and M. Kurimo. Importance of high-order n-gram models in morph-based speech recognition. *IEEE Trans. Audio, Speech and Language Processing*, 17(4):724–732, May 2009.

T. Hori, I. L. Hetherington, T. J. Hazen, and J. R. Glass. Open-vocabulary spoken utterance retrieval using confusion networks. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processin (ICASSP'07)*, volume 4, pages 73–76, Apr. 2007.

E. Hyvönen, E. Mäkelä, M. Salminen, A. Valo, K. Viljanen, S. Saarela, M. Junnila, and S. Kettula. MuseumFinland–Finnish museums on the semantic web. *Web Semantics: Science, Services and Agents on the World Wide Web*, 3(2–3): 224–241, 2005. ISSN 1570-8268. Selcted Papers from the International Semantic Web Conference.

M. Jalanko. *Studies of learning projective methods in automatic speech recognition*. PhD thesis, Helsinki University of Technology, 1980.

D. James. *The Application of Classical Information Retrieval Techniques to Spoken Documents*. PhD thesis, University of Cambridge, UK, 1995.

D. James and S. Young. A fast lattice-based approach to vocabulary independent wordspotting. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'94)*, volume 1, pages 377–380, Apr. 1994.

K. Järvelin, J. Kekäläinen, and T. Niemi. ExpansionTool: Concept-based query expansion and construction. *Information Retrieval*, 4(3):231–255, 2001. ISSN 1386-4564.

F. Jelinek and R. L. Mercer. Interpolated estimation of Markov source parameters from sparse data. In *Proceedings of the Workshop on Pattern Recognition in Practice*, pages 381–397, Amsterdam, The Netherlands, May 1980.

S. Johnson, P. Jourlin, G. Moore, K. Spärck Jones, and P. C. Woodland. The cambridge university spoken document retrieval system. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'99)*, volume 1, pages 49–52, Phoenix, AZ, 1999.

G. J. F. Jones, J. T. Foote, K. Spärck Jones, and S. J. Young. Retrieving spoken documents by combining multiple index sources. In *Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 30–38, New York, NY, USA, 1996. ACM. ISBN 0-89791-792-8.

P. Jourlin, S. E. Johnson, K. Spärck Jones, and P. C. Woodland. General Query Expansion Techniques for Spoken Document Retrieval. In *Proceedings of ESCA Workshop on Extracting Information from Spoken Audio*, pages 8–13, Cambridge, UK, 1999.

F. Karlsson and K. Koskenniemi. A process model of morphology and lexicon. *Folia Linguistica*, 19(1–2):207–231, 1985.

S. Kaski, T. Honkela, K. Lagus, and T. Kohonen. Websom – self-organizing maps of document collections. *Neurocomputing*, 21(1–3):101–117, 1998. ISSN 0925-2312.

S. M. Katz. Estimation of probabilities from sparse data for the language model component of a speech recognizer. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 35(3):400–401, Mar. 1987.

J. Kekäläinen and K. Järvelin. Using graded relevance assessments in IR evaluation. *J. Am. Soc. Inf. Sci. Technol.*, 53(13):1120–1129, 2002.

R. Kneser and H. Ney. Improved backing-off for m-gram language modeling. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'95)*, pages 181–184, 1995.

O. Kohonen, S. Virpioja, and M. Klami. Allomorfessor: Towards unsupervised morpheme analysis. In *Evaluating systems for Multilingual and Multimodal Information Access, 9th Workshop of the Cross-Language Evaluation Forum, CLEF 2008, Aarhus, Denmark, September 17-19, 2008, Revised Selected Papers*, volume 5706 of *Lecture Notes in Computer Science*. Springer, 2009.

K. Koskenniemi. *Two-Level Morphology: A General Computational Model for Word-Form Recognition and Production*. PhD thesis, University of Helsinki, 1983.

M. Kurimo. *Using Self-Organizing Maps and Learning Vector Quantization for Mixture Density Hidden Markov Models*. PhD thesis, Helsinki University of Technology, Espoo, Finland, 1997.

M. Kurimo. Fast latent semantic indexing of spoken documents by using self-organizing maps. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing ICASSP'00*, volume 4, pages 2425–2428, June 2000.

M. Kurimo, V. Turunen, and I. Ekman. Speech transcription and spoken document retrieval in Finnish. In *In Machine Learning for Multimodal Interaction, Revised Selected Papers of the MLMI 2004 workshop*, Lecture Notes in Computer Science, vol. 3361, pages 253–262. Springer, 2005.

M. Kurimo, M. Creutz, and K. Lagus. Unsupervised segmentation of words into morphemes - challenge 2005, an introduction and evaluation report. In *Proceedings of the PASCAL Challenge Workshop on Unsupervised Segmentation of Words into Morphemes*, pages 1–11, Venice, Italy, 2006a. PASCAL European Network of Excellence.

M. Kurimo, A. Puurula, E. Arisoy, V. Siivola, T. Hirsimäki, J. Pylkkönen, T. Alumäe, and M. Saraclar. Unlimited vocabulary speech recognition for agglutinative languages. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, HLT-NAACL '06, pages 487–494, Stroudsburg, PA, USA, 2006b. ACL.

M. Kurimo, M. Creutz, and M. Varjokallio. Morpho Challenge evaluation using a linguistic Gold Standard. In *Advances in Multilingual and Multimodal Information Retrieval, 8th Workshop of the Cross-Language Evaluation Forum, CLEF 2007, Budapest, Hungary, September 19-21, 2007, Revised Selected Papers*, volume 5152 of *Lecture Notes in Computer Science*, pages 864–873. Springer, 2008.

M. Kurimo, M. Creutz, and V. Turunen. Morpho Challenge evaluation by information retrieval experiments. In C. Peters, T. Deselaers, N. Ferro, J. Gonzalo, G. Jones, M. Kurimo, T. Mandl, A. Peñas, and V. Petras, editors, *Evaluating Systems for Multilingual and Multimodal Information Access*, volume 5706 of

Lecture Notes in Computer Science, pages 991–998. Springer Berlin / Heidelberg, 2009a. ISBN 978-3-642-04446-5.

M. Kurimo, V. Turunen, and M. Varjokallio. Overview of Morpho Challenge 2008. In C. Peters, T. Deselaers, N. Ferro, J. Gonzalo, G. Jones, M. Kurimo, T. Mandl, A. Peñas, and V. Petras, editors, *Evaluating Systems for Multilingual and Multimodal Information Access*, volume 5706 of *Lecture Notes in Computer Science*, pages 951–966. Springer Berlin / Heidelberg, 2009b. ISBN 978-3-642-04446-5.

M. Kurimo, S. Virpioja, and V. T. Turunen. Overview and results of Morpho Challenge 2010. In *Proceedings of the Morpho Challenge 2010 Workshop*, pages 7–24, Espoo, Finland, Sept. 2010a. Aalto University School of Science and Technology, Department of Information and Computer Science. Technical Report TKK-ICS-R37.

M. Kurimo, S. Virpioja, V. T. Turunen, G. W. Blackwood, and W. Byrne. Overview and results of Morpho Challenge 2009. In *Multilingual Information Access Evaluation I. Text Retrieval Experiments: 10th Workshop of the Cross-Language Evaluation Forum, CLEF 2009, Corfu, Greece, September 30 - October 2, 2009, Revised Selected Papers*, volume 6241 of *Lecture Notes in Computer Science*, pages 578–597. Springer, 2010b.

J. Laaksonen, M. Koskela, S. Laakso, and E. Oja. Picsom – content-based image retrieval with self-organizing maps. *Pattern Recognition Letters*, 21(13–14): 1199–1207, 2000. ISSN 0167-8655.

M. Larson, S. Eickeler, and J. Kohler. Supporting radio archive workflows with vocabulary independent spoken keyword search. In *Proceedings of the ACM SIGIR Workshop on Searching Spontaneous Conversational Speech*, pages 15–22, Amsterdam, The Netherlands, 2007.

M. Lehtonen and A. Doucet. XML-aided phrase indexing for hypertext documents. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 843–844, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-164-4.

B. Liu and D. W. Oard. One-sided measures for evaluating ranked retrieval effectiveness with spontaneous conversational speech. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 673–674, New York, NY, USA, 2006. ACM. ISBN 1-59593-369-7.

B. Logan and J. V. Thong. Confusion-based query expansion for OOV words in spoken document retrieval. In *Proceedings of 7th International Conference on Spoken Language Processing (ICSLP2002 – Interspeech 2002)*, pages 1997–2000, Sept. 2002.

B. Logan, P. Moreno, and O. Deshmukh. Word and sub-word indexing approaches for reducing the effects of OOV queries on spoken audio. In *Proceedings of the second international conference on Human Language Technology Research*, HLT '02, pages 31–35, San Francisco, CA, USA, 2002. Morgan Kaufmann Publishers Inc.

I. Malioutov and R. Barzilay. Minimum cut model for spoken lecture segmentation. In *Proceedings of the 21st International Conference on Computational*

Linguistics and 44th Annual Meeting of the Association for Computational Linguistics, pages 25–32, Sydney, Australia, July 2006. Association for Computational Linguistics.

J. Mamou, D. Carmel, and R. Hoory. Spoken document retrieval from call-center conversations. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 51–58, New York, NY, USA, 2006. ACM Press. ISBN 1-59593-369-7.

J. Mamou, B. Ramabhadran, and O. Siohan. Vocabulary independent spoken term detection. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 615–622, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-597-7.

L. Mangu, E. Brill, and A. Stolcke. Finding consensus in speech recognition: word error minimization and other applications of confusion networks. *Computer Speech & Language*, 14(4):373–400, 2000.

C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA, 2008. ISBN 0521865719, 9780521865715.

M. Markkula and E. Sormunen. End-user searching challenges indexing practices in the digital newspaper photo archive. *Information Retrieval*, 1(4):259–285, 2000. ISSN 1386-4564.

P. McNamee and J. Mayfield. N-gram morphemes for retrieval. In *Working Notes for the CLEF 2007 Workshop*, Budapest, Hungary, Sept. 2007.

T. Mertens and D. Schneider. Efficient subword lattice retrieval for german spoken term detection. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'09)*, pages 4885–4888, Apr. 2009.

M. Mohri, F. Pereira, and M. Riley. Weighted finite-state transducers in speech recognition. *Computer Speech & Language*, 16(1):69–88, 2002. ISSN 0885-2308.

C. Monson, J. Carbonell, A. Lavie, and L. Levin. ParaMor: Finding paradigms across morphology. In *Advances in Multilingual and Multimodal Information Retrieval, 8th Workshop of the Cross-Language Evaluation Forum, CLEF 2007, Budapest, Hungary, September 19-21, 2007, Revised Selected Papers*, volume 5152 of *Lecture Notes in Computer Science*, pages 900–907. Springer, 2008.

N. Morgan, D. Baron, J. Edwards, D. Ellis, D. Gelbart, A. Janin, T. Pfau, E. Shriberg, and A. Stolcke. The meeting project at ICSI. In *Proceedings of the first international conference on Human language technology research*, HLT'01, pages 1–7, Stroudsburg, PA, USA, 2001. ACL.

J. Morris and G. Hirst. Lexical cohesion computed by thesaural relations as an indicator of the structure of text. *Comput. Linguist.*, 17(1):21–48, Mar. 1991. ISSN 0891-2017.

H. Mourujärvi. Radio and television archive opened for public use. *CSC News*, 15(1):25–27, 2011. ISSN 1239-9248. URL `http://www.csc.fi/english/csc/publications/cscnews/2011/1/`.

P. V. Mulbregt, I. Carp, L. Gillick, S. Lowe, and J. Yamron. Text segmentation and topic tracking on broadcast news via a hidden Markov model approach. In *Proceedings of the 5th International Conference on Spoken Language Processing (ICSLP'98)*, pages 2519–2522, 1998.

K. Ng and V. W. Zue. Subword-based approaches for spoken document retrieval. *Speech Communication*, 32(3):157–186, 2000. ISSN 0167-6393.

S. Nießen and H. Ney. Statistical machine translation with scarce resources using morpho-syntactic information. *Computational Linguistics*, 30(2):181–204, 2004.

A. Olney and Z. Cai. An orthonormal basis for topic segmentation in tutorial dialogue. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, pages 971–978, Stroudsburg, PA, USA, 2005. ACL.

J. S. Olsson. *Combining Evidence from Unconstrained Spoken Term Frequency Estimation for Improved Speech Retrieval*. PhD thesis, University of Maryland, 2008.

J. S. Olsson and D. W. Oard. Combining LVCSR and vocabulary-independent ranked utterance retrieval for robust speech search. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 91–98, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-483-6.

Y. Pan and L. Lee. Performance analysis for lattice-based speech indexing approaches using words and subword units. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(6):1562–1574, Aug. 2010. ISSN 1558-7916.

Y. C. Pan, H. L. Chang, and L. S. Lee. Analytical comparison between position specific posterior lattices and confusion networks based on words and subword units for spoken document indexing. In *Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pages 677–682, Kyoto, Japan, 2007.

S. Parlak and M. Saraclar. Spoken term detection for turkish broadcast news. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'08)*, pages 5244–5247, Apr. 2008.

R. J. Passonneau and D. J. Litman. Discourse segmentation by human and automated means. *Comput. Linguist.*, 23(1):103–139, 1997. ISSN 0891-2017.

P. Pecina, P. Hoffmannová, G. Jones, Y. Zhang, and D. Oard. Overview of the CLEF-2007 cross-language speech retrieval track. In C. Peters, V. Jijkoun, T. Mandl, H. Müller, D. Oard, A. Peñas, V. Petras, and D. Santos, editors, *Advances in Multilingual and Multimodal Information Retrieval*, volume 5152 of *Lecture Notes in Computer Science*, pages 674–686. Springer Berlin / Heidelberg, 2008. ISBN 978-3-540-85759-4.

A. Pirkola, T. Hedlund, H. Keskustalo, and K. Järvelin. Dictionary-based cross-language information retrieval: Problems, methods, and research findings. *Information Retrieval*, 4(3):209–230, 2001. ISSN 1386-4564.

J. M. Ponte and W. B. Croft. A language modeling approach to information retrieval. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 275–281, New York, NY, USA, 1998. ACM. ISBN 1-58113-015-5.

M. F. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.

K. Puolamäki, J. Salojärvi, E. Savia, J. Simola, and S. Kaski. Combining eye movements and collaborative filtering for proactive information retrieval. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 146–153, New York, NY, USA, 2005. ACM. ISBN 1-59593-034-5.

A. Puurula and M. Kurimo. Vocabulary decomposition for estonian open vocabulary speech recognition. In *45th Annual Meeting of the Association for Computational Linguistics*, pages 89–95. ACL, 2007.

J. Pylkkönen. An efficient one-pass decoder for finnish large vocabulary continuous speech recognition. In *Proceedings of Second Baltic Conference on Human Language Technologies*, pages 167–172, 2005.

J. Pylkkönen. Investigations on discriminative training in large scale acoustic model estimation. In *Proceedings of 10th Annual Conference of the International Speech Communication Association (Interspeech 2009)*, pages 220–223, 2009.

J. Pylkkönen and M. Kurimo. Using phone durations in Finnish large vocabulary continuous speech recognition. In *Proceedings of the 6th Nordic Signal Processing Symposium (Norsig)*, pages 324–326, 2004.

L. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, Feb. 1989. ISSN 0018-9219.

U. Remes, K. Palomaki, T. Raiko, A. Honkela, and M. Kurimo. Missing-feature reconstruction with a bounded nonlinear state-space model. *IEEE Signal Processing Letters*, 18(10):563–566, Oct. 2011. ISSN 1070-9908.

S. Renals and D. Ellis. Audio information access from meeting rooms. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'03)*, volume 4, pages 744–747, Apr. 2003.

S. Renals, D. Abberley, D. Kirby, and T. Robinson. Indexing and retrieval of broadcast news. *Speech Communication*, 32(1–2):5–20, 2000.

J. C. Reynar. An automatic method of finding topic boundaries. In *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, ACL '94, pages 331–333, Stroudsburg, PA, USA, 1994. ACL.

F. Richardson, M. Ostendorf, and J. Rohlicek. Lattice-based search strategies for large vocabulary speech recognition. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'95)*, volume 1, pages 576–579, May 1995.

J. Rissanen. *Stochastic Complexity in Statistical Inquiry*. World Scientific, River Edge, New Jersey, 1989.

S. E. Robertson. On term selection for query expansion. *Journal of Documentation*, 46(4):359–364, Jan. 1990. ISSN 0022-0418.

S. E. Robertson and K. Spärck Jones. Relevance weighting of search terms. *Journal of the American Society for Information Science*, 27(3):129–146, 1976. ISSN 1097-4571.

S. E. Robertson, S. Walker, S. Jones, M. Hancock-Beaulieu, and M. Gatford. Okapi at TREC-3. In D. Harman, editor, *Overview of the Third Text REtrieval Conference (TREC-3)*, pages 109–126, Gaithersburg, MD, 1995. NIST.

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

G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, 24(5):513–523, 1988.

G. Salton, A. Wong, and C. S. Yang. A vector space model for automatic indexing. *Commun. ACM*, 18(11):613–620, 1975. ISSN 0001-0782.

M. Saraçlar and R. Sproat. Lattice-based search for spoken utterance retrieval. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HTL-NAACL)*, pages 129–136, 2004.

E. Shriberg, A. Stolcke, D. Hakkani-Tür, and G. Tür. Prosody-based automatic segmentation of speech into sentences and topics. *Speech Communication*, 32 (1–2):127–154, 2000. ISSN 0167-6393.

M. A. Siegler. *Integration of Continuous Speech Recognition and Information Retrieval for Mutually Optimal Performance*. PhD thesis, Carnegie Mellon University, 1999.

V. Siivola. *Induction of the Morphology of Natural Language: Unsupervised Morpheme Segmentation with Application to Automatic Speech Recognition*. PhD thesis, Helsinki University of Technology, 2006.

V. Siivola, T. Hirsimäki, M. Creutz, and M. Kurimo. Unlimited vocabulary speech recognition based on morphs discovered in an unsupervised manner. In *Proceedings of the 8th European Conference on Speech Communication and Technology (Eurospeech 2003 - Interspeech 2003)*, pages 2293–2296, 2003.

V. Siivola, T. Hirsimäki, and S. Virpioja. On growing and pruning Kneser-Ney smoothed n-gram models. *IEEE Transactions on Audio, Speech & Language Processing*, pages 1617–1624, 2007.

A. Singhal, C. Buckley, and M. Mitra. Pivoted document length normalization. In *Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '96, pages 21–29, New York, NY, USA, 1996. ACM. ISBN 0-89791-792-8.

M. Sjöberg, S. Ishikawa, M. Koskela, J. Laaksonen, and E. Oja. PicSOM experiments in TRECVID 2011. In *Proceedings of the TRECVID 2011 Workshop*, Gaithersburg, MD, USA, Dec. 2011.

K. Spärck Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28(1):11–21, 1972.

S. Spiegler and C. Monson. EMMA: A novel evaluation metric for morphological analysis. In *Proceedings of the 23rd International Conference on Computational Linguistics (COLING)*, pages 1029–1037, Stroudsburg, PA, USA, Aug. 2010. ACL.

K. Torkkola. *Short-time feature vector based phonemic speech recognition with the aid of local context*. PhD thesis, Helsinki University of Technology, 1991.

G. Tür, D. Hakkani-Tür, A. Stolcke, and E. Shriberg. Integrating prosodic and lexical cues for automatic topic segmentation. *Computational Linguistics*, 27 (1):31–57, Mar. 2001.

C. J. van Rijsbergen. *Information Retrieval*. Butterworth, 1979. ISBN 0-408-70929-4.

E. M. Voorhees. Query expansion using lexical-semantic relations. In *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '94, pages 61–69, New York, NY, USA, 1994. Springer-Verlag New York, Inc. ISBN 0-387-19889-X.

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

E. Whittaker, J. Van Thong, and P. Moreno. Vocabulary independent speech recognition using particles. In *Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pages 315–318, 2001.

P. Woodland, C. Leggetter, J. Odell, V. Valtchev, and S. Young. The 1994 HTK large vocabulary speech recognition system. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'95)*, volume 1, pages 73–76, May 1995. ISBN 0-7803-2431-5.

P. C. Woodland, S. E. Johnson, P. Jourlin, and K. Spärck Jones. Effects of out of vocabulary words in spoken document retrieval. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 372–374, New York, NY, USA, 2000. ACM. ISBN 1-58113-226-3.

B. Xiang, K. Nguyen, L. Nguyen, R. Schwartz, and J. Makhoul. Morphological decomposition for Arabic broadcast news transcription. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing ICASSP'06*, volume 1, pages 1089–1092, May 2006.

S. J. Young, N. H. Russell, and J. H. S. Thornton. Token passing: A simple conceptual model for connected speech recognition systems. Technical Report CUED/F-INFENG/TR38, Cambridge University Engineering Dept., 1989.

S. J. Young, J. J. Odell, and P. C. Woodland. Tree-based state tying for high accuracy acoustic modelling. In *Proceedings of the workshop on Human Language Technology*, HLT '94, pages 307–312, Stroudsburg, PA, USA, 1994. ACL. ISBN 1-55860-357-3.

YouTube. Statistics, 2012. URL `http://www.youtube.com/t/press_statistics`. Accessed May 14, 2012.

P. Yu, K. Chen, C. Ma, and F. Seide. Vocabulary-independent indexing of spontaneous speech. *IEEE Transactions on Speech and Audio Processing*, 13(5): 635–643, Sept. 2005. ISSN 1063-6676.

J. Zheng and A. Stolcke. Improved discriminative training using phone lattices. In *Proceedings of the 9th European Conference on Speech Communication and Technology (Interspeech 2005–Eurospeech)*, pages 2125–2128, Sept. 2005.

DISSERTATIONS IN INFORMATION AND COMPUTER SCIENCE

Aalto-DD99/2011 Ojala, Markus
Randomization Algorithms for Assessing the Significance of Data Mining Results. 2011.

Aalto-DD111/2011 Dubrovin, Jori
Efficient Symbolic Model Checking of Concurrent Systems. 2011.

Aalto-DD118/2011 Hyvärinen, Antti
Grid Based Propositional Satisfiability Solving. 2011.

Aalto-DD136/2011 Brumley, Billy Bob
Covert Timing Channels, Caching, and Cryptography. 2011.

Aalto-DD11/2012 Vuokko, Niko
Testing the Significance of Patterns with Complex Null Hypotheses. 2012.

Aalto-DD19/2012 Reunanen, Juha
Overfitting in Feature Selection: Pitfalls and Solutions. 2012.

Aalto-DD33/2012 Caldas, José
Graphical Models for Biclustering and Information Retrieval in Gene Expression Data. 2012.

Aalto-DD45/2012 Viitaniemi, Ville
Visual Category Detection: an Experimental Perspective. 2012.

Aalto-DD51/2012 Hanhijärvi, Sami
Multiple Hypothesis Testing in Data Mining. 2012.

Aalto-DD56/2012 Ramkumar, Pavan
Advances in Modeling and Characterization of Human Neuromagnetic Oscillations. 2012

Large amounts of video and audio material are produced, distributed, and stored every day. Speech retrieval techniques combine automatic speech recognition and information retrieval, and allow users to type in search terms and find the matching segments based on the speech content. However, the vocabulary of the speech recognizer is limited and the rarest words in the language can never be recognized correctly. This is problematic for retrieval, because search terms are often rare words such as proper names. This thesis proposes a morpheme based approach, where any word can be retrieved by first recognizing its component morphemes. Further improvements can be made by using alternative recognition candidates, and by inferring semantic relations of index terms.

Aalto-DD 97/2012

ISBN 978-952-60-4717-1
ISBN 978-952-60-4718-8 (pdf)
ISSN-L 1799-4934
ISSN 1799-4934
ISSN 1799-4942 (pdf)

Aalto University
School of Science
Department of Information and Computer Science
www.aalto.fi

BUSINESS + ECONOMY

ART + DESIGN + ARCHITECTURE

SCIENCE + TECHNOLOGY

CROSSOVER

DOCTORAL DISSERTATIONS