Kirkedal, Andreas Søeborg

**Doctoral Thesis**

# Danish Stød and Automatic Speech Recognition

PhD Series, No. 24.2016

**Provided in Cooperation with:**
Copenhagen Business School (CBS)

*Suggested Citation:* Kirkedal, Andreas Søeborg (2016) : Danish Stød and Automatic Speech Recognition, PhD Series, No. 24.2016, ISBN 9788793483132, Copenhagen Business School (CBS), Frederiksberg,
https://hdl.handle.net/10398/9336

This Version is available at:
https://hdl.handle.net/10419/208978

**COPENHAGEN BUSINESS SCHOOL**
SOLBJERG PLADS 3
DK-2000 FREDERIKSBERG
DANMARK

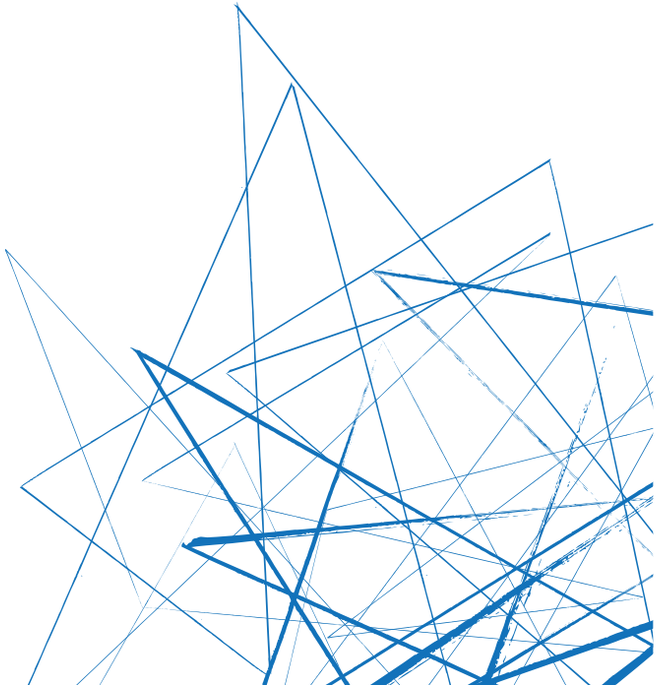
**WWW.CBS.DK**

**ISSN 0906-6934**

**Print ISBN: 978-87-93483-12-5**
**Online ISBN: 978-87-93483-13-2**

**DANISH STØD AND AUTOMATIC SPEECH RECOGNITION**

PhD Series 24-2016



**Andreas Søeborg Kirkedal**

# DANISH STØD AND AUTOMATIC SPEECH RECOGNITION

The PhD School of LIMAC

**PhD Series 24.2016**

CBS COPENHAGEN BUSINESS SCHOOL
HANDELSHØJSKOLEN

# Danish Stød and Automatic Speech Recognition

Andreas Søeborg Kirkedal

Industrial Ph.D. collaboration with Mirsk Digital ApS

Academic advisor: Associate Professor Peter Juul Henrichsen, Ph.D.

Ph.D. School LIMAC, Programme for Language, Culture and Communication

Copenhagen Business School

Andreas Søeborg Kirkedal
*Danish Stød and Automatic Speech Recognition*

1st edition 2016
PhD Series 24.2016



ISSN 0906-6934

Print ISBN: 978-87-93483-12-5
Online ISBN: 978-87-93483-13-2

LIMAC PhD School is a cross disciplinary PhD School connected to research communities within the areas of Languages, Law, Informatics, Operations Management, Accounting, Communication and Cultural Studies.

# Preface

This thesis presents original research that was submitted to fulfill the requirements to obtain the degree *Philosophiae Doctor* (Ph.D.) in the topics natural language processing and speech technology.

The present thesis is part of an Industrial Ph.D. project that was a collaboration between Danish Centre for Applied Speech Technology (DanCAST), Department of International Business Communication, Copenhagen Business School and Mirsk Digital ApS, Copenhagen, Denmark and supported by the Danish Agency for Science, Technology and Innovation.

During the project life-time from September 2012 to May 2016, a methodology to create speech recognisers for Danish and several speech recognition systems were developed as well as three published papers and two papers accepted for publication.

The presented research was supervised by Associate Professor Peter Juel Henrichsen (DanCAST), Department of International Business Communication, Copenhagen Business School, and CTO Klaus Akselsen, Mirsk Digital ApS, Denmark.

Andreas Søeborg Kirkedal
July 16, 2016
Islands Brygge, Denmark

## Abstract

Stød is a prosodic feature in Danish spoken language that is able to distinguish lexemes. This distinction can also identify word class and has the potential to improve the performance of automatic speech recognisers for Danish spoken language. Stød manifestation exhibits a large amount of variability and may be perceptual in nature, because stød in some cases can be audibly perceived yet not be visible in a spectrogram. The variability is the primary reason there is currently no agreed upon acoustic or phonetic definition of stød. The working definition of stød is "*... a kind of creaky voice*" (Grønnum, 2005) and "*stød is not* just *creak*" (Hansen, 2015).

In the present work, we investigate whether stød can be exploited in automatic speech recognition. To exploit stød without an acoustic or phonetic definition, we need to use a (almost) zero-knowledge data-driven approach which is based on a number of assumptions that we investigate prior to conducting ASR experimentation. We assume that stød can be detected in audio input, using *acoustic features*. To detect stød, we need to identify features that signal stød, which requires *annotated data*. To select the right features, the *stød annotation* must be reliable and accurate.

We therefore conduct a reliability study of stød annotation with inter-annotator agreement measures, rank acoustic features for stød detection according to feature importance using a forest of randomised decision trees and experiment with stød detection as a binary and multi-class classification task. The experiments identify a set of features important or stød detection and confirms that we can detect stød in audio.

Lastly, we model stød in automatic speech recognition and show that significant improvements in word error rate can be gained simply by annotating stød in the phonetic dictionary at the expense of decoding speed. Extending the acoustic feature vectors with pitch-related features and other features of voice quality also give significant performance improvement on both read-aloud speech and spontaneous speech. Decoding speed increases when we extend the acoustic feature vectors and actually improve decoding speed over the baseline where stød is not modelled.

## Resumé

Stød er en kontrastiv prosodi i dansk talesprog, som er betydningsadskillende. Af denne årsag antages det, at automatisk genkendelse af dansk talesprog kan forbedres, hvis den kan tage højde for stød. Stødrealisering udviser stor variabilitet i akustisk analyse og er derfor et svært definerbart fænomen. Stød beskrives som regel som "*... en art knirkestemme*"(Grønnum, 2005) og "*stød er ikke* kun *knirk*" (Hansen, 2015).

I denne afhandling undersøges hvorvidt stød kan bruges til at forbedre automatisk talegenkendelse. Da der ikke findes en dækkende fonetisk eller akustisk definition af stødet, vil vi bruge en data-dreven tilgang til undesøgelsen, som baserer sig på en række antagelser, der skal undersøges inden stødet kan integreres i talegenkendelsessystemer. En antagelse er, at vi kan detektere stød i akustiske mål. For at detektere stød skal vi identificere de akustiske mål, der signalerer stød, hvilket kræver at vi har adgang til data, der er annoteret med stød. Annoteringen af stød skal være pålidelig, hvis analysen af akustiske mål skal være retvisende.

Hvis disse antagelser viser sig at være korrekte, kan vi estimere statistiske modeller som detekterer stød i akustisk input. Hvis modellerne kan forudsige stød med tiltrækkelig nøjagtighed, kan stødinformation tilføjes til det akustiske input, dvs. vektorer af akustiske mål, der bruges som input til talegenkendelse, og til talegenkenderens fonetiske ordbog.

For at undersøge vores antagelser om stødet foretager vi en pålideligheds-undersøgelse af stødannotering. Derefter udtrækker vi 120 akustiske mål, som vi rangerer baseret på deres evne til at signalere stødforekomst. Denne rangering anvender vi til at udvælge specifikke akustiske mål til at estimere statistiske modeller, der detekterer stød i lyd. Vi identificerer 17 akustiske mål som signalerer stød og bekræfter at vi kan detektere stød i akustiske mål.

Bevæbnet med denne viden integreres stød i talegenkendelse, og vi påviser at man kan opnå signifikant bedre talegenkendelse på bekostning af genkendelseshastighed, hvis stød annoteres i den fonetiske ordbog. Ved at tilføje akustiske mål for stemmekvalitet, som blev rangeret højt for deres evne til at signalere stød, til talegenkendelsesinput opnås signifikant forbedret talegenkendelse af både oplæst tale og spontantale fra tre forskellige datasæt, hvilket samtidig kompenserer for den nedsatte genkendelseshastighed.

## Contribution

The specific contributions to the understanding of stød and its use in automatic speech recognition in this thesis are listed below in bullet form:

1. Expert stød annotation is reliable
2. 18 features carry information that signal stød: the first four 4 MFCC and PLP features, Probability-of-Voicing, Log-pitch, Peak Slope, Harmonic Richness Factor, and the phase features Phase Distortion Mean 13-14 and Phase Distortion Deviation 10-13
3. Stød can be detected in acoustic features when stød is predicted jointly with the underlying segment
4. ASR systems that model stød can significantly outperform corresponding systems that do not, if the ASR systems are trained on LDA-projected MFCC features
5. Extending MFCC feature vectors with Probability-of-Voicing, Log-pitch, and the Harmonic Richness Factor or the Phase Distortion Mean features 13 & 14 and Phase Distortion Deviation 10-13 improve both word error rate and decoding speed for ASR systems that model stød
6. The first freely available ASR system for Danish spoken language that includes methodology and data

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

Automatic speech recognition (ASR) denotes the complicated process of translating spoken language to written language. ASR in general performs at sub-human levels and ASR for Danish suffers from lack of data and software tools which have resulted in a sparse amount of research in the area (Pedersen et al., 2012). Speech has been a popular input modality for electronic devices for several years in a number of domains and applications, from automated telephone customer services to legal and clinical documentation and ASR performance becomes vital if speech is used to interface with more and more devices. If voice control using Danish performs poorly, Danes will shift to English instead and accelerate anglicisation. The future of Danish as a digital language looks brighter if Danish spoken language can be used to interface with the multitude of electronic devices, such as wrist watches or glasses, which are being developed and are too small to control using keyboard or mouse.

The purpose of the work conducted in this thesis is to improve, stimulate and advance research and development of Danish ASR and is intended for researchers in linguistics, natural language processing (NLP), and developers of speech technology. We hypothesise that recognition rates of Danish large-vocabulary ASR can improve by modelling the Danish prosodic feature *stød* in ASR systems.

Danish stød was first described in Høysgaard (1743) and has been treated by many researchers in the linguistic community. This has resulted in substantial number of scientific articles on Danish stød[1] and stød is known in phonetics around the globe (Bőhm, 2009; Jurgec, 2007; Frazier, 2013). Stød is interesting in Danish spoken language for several reasons:

1. Stød is a *distinctive* distinctive feature. Stød can be the only feature that distinguishes lexical items.

[1] See Section 2.2.1 for references.

2. Stød is a *prosodic* feature. Prosody affects the sounds or phones that are used to utter a word, but stød is not a sound in itself.

3. Stød is a *perceptual* feature. When informants hear words that are minimal pairs where the distinctive feature is stød, the subjects can identify the lexical item from the utterance with high accuracy, but it is difficult to identify the acoustic marker signalling the presence of stød.

Because stød is distinctive, it is very useful to detect stød from acoustic input. Some words that are distinguished by stød, e.g. *viser* (noun, EN: hand on a clock) vs. *viser* (verb, EN: to show) and *maler* (noun, EN: a painter) vs. *maler* (verb, EN: to paint) are homographs, but many words pairs are not. Examples are *mand* (noun, EN: a man) vs. *man* (pronoun, EN: you/one)Stød is considered a perceptual feature because stød can be audibly heard by a listener, but be realised very differently by speakers or be hardly visible in acoustic analysis (Hansen, 2015).

If stød detection from acoustic input is possible, the added annotation at the phonetic level will distinguish several minimal pairs. If word pairs that would otherwise be identical can be distinguished by stød, a speech recogniser is more likely to recognise the correct word.

Currently, speech recognisers use syntax to choose the most likely word. Syntax is learned from tens or hundreds of millions of running words and data sets of this size are available in some major languages such as English, but not generally in Danish, especially for specialised domains such as medical dictation. There are scenarios where stød is the only distinguishing feature and the lack of powerful syntax models can be alleviated if the distinctive lexical function of stød can be recognised directly from the acoustic signal.

## 1.1 Potential

The largest consumer of large-vocabulary ASR in Denmark is the medical sector. A recent study found that medical secretaries use an average of 7.8 hours per week on transcription (Implement, 2009). The clinical documentation workflow itself will add to that figure (how much depends on the implementation), but transcription itself accounts for approximately 1/5 of their workload. Reducing the transcription workload using ASR can potentially free a significant amount of resources.

In the 1990s and 2000s, digital dictation systems became available for medical dictation and the Danish government decided to digitise all medical records in one national electronic medical records system, which is part of a large-scale effort to digitise administration in the Danish public sector. A national electronic medical records system improves documentation, accessibility and performance measurement, and makes it possible to access a medical record from multiple places at the same time, e.g. at a doctor's conference, where

the treatment is discussed, and in the emergency room at a medical emergency happening simultaneously. If a patient is admitted to a hospital, the patient record is immediately available and need not be retrieved from his/her general practitioner, etc.

Working with both patient records and transcriptions on a computer provide an improved and more efficient workflow and a scenario well-suited for ASR-augmentation. A report on the efficiency gains achievable with digital dictation reports that 22.4% more dictations were processed per secretary each day. For a clinic with 20 physicians and 10 medical secretaries, as much as 1963 staff hours per year could be gained from increased efficiency due to digital dictation (Barsøe Management, 2008).

While a national electronic medical records system has not been fully implemented in Denmark due to difference in documentation across regions, hospitals and specialisations, electronic medical records systems have been implemented in all hospitals and many clinics today and use standardised exchange formats. The challenges faced by hospitals today are (Gjørup, 2010):

1. Medical records are not available nor are they up-to-date for the physician responsible for patient care.
2. Clinical decisions can be based on incomplete, outdated or wrong information.
3. Some clinical decisions are postponed or not made.
4. Patient safety is compromised because the patient is sent to treatment before the medical record is completed.

In other words, transcription is still slow. The missing or incomplete medical records may also lead to a negative spiral with respect to time usage because secretaries and physicians take extra time to find the information. Some hospitals have tried to use transcription agencies to manage the workload and free resources, but external transcribers do not understand medical terminology and transcription accuracy decreases due to misapprehension. Recent legislation also requires that medical records can be accessed by the patient in question and that the attending physician approves the transcription before a diagnosis or treatment is documented in the electronic patient record. This makes the use of external transcription agencies problematic from a workflow perspective, because the attending physician will not be able to approve the transcription. If a transcription is finished days or months after consultation or operation, the physician will have no recollection of the specifics of the diagnosis. Even if the transcription is finished later the same day, the physician will have conducted several patient consultations in the meantime and specifics, such as whether an operating room should be booked or which medicine has been subscribed, may have been forgotten.

### 1.1.1 Medical dictation and speech recognition

To make transcription more efficient, a significant amount of work has been devoted to augment medical dictation with ASR. Medical dictation is characterised by free text documentation which means that large volumes of running text is produced every day and that full natural language is used in the documentation. Speech-enabled interfaces have been are proven to be more effective than keyboard-and-mouse interfaces for tasks where full natural language communication is useful or where keyboard and mouse are not appropriate (Jurafsky & Martin, 2008). Medical dictation is also a natural area to apply ASR-augmentation because dictation is intended to produce written text. The data usually used to train ASR systems is read-aloud text because reliable transcriptions are available. Dictation is in a sense the reverse process of reading aloud. Dictation is not as structured as read-aloud text, but has more structure than spontaneous speech.

Though there are an abundance of text in medical dictation that could be used for statistical language modelling, it can be difficult to acquire in-domain training data due to the sensitive information contained therein. It is therefore desirable to achieve the best possible speech recognition output by utilising information in the speech signal and that makes stød an attractive feature to investigate from a commercial point of view.

There are two scenarios in medical dictation where ASR can remove or alleviate the problems mentioned above: Real-time ASR and ASR+post-editing.

#### Real-time automatic speech recognition

Speaking is faster than typing (Basapur et al., 2007). If the physician uses digital dictation augmented with real-time ASR, the secretary is not a part of the documentation workflow and a resource is free for other purposes. As a side-effect, the physician is the last eyes on the transcription and can approve or correct a transcription immediately while the consultation is still fresh in memory. If integrated with an electronic medical records system, the physician can even dictate directly into the patient record and the clinical documentation will always be up-to-date with the most recent information.

#### Automatic speech recognition and post-editing

In the earliest efforts in NLP, ASR was expected to completely replace – rather than enhance – other input modes. However, speech input achieves better performance in combination with other input modalities for many tasks (Pausch & Leatherby, 1991). High accuracy, real-time ASR is necessary to realise the potential efficiency gains sought by hospitals. If ASR accuracy is not high enough, the physician will spend time post-editing the ASR output. While this still frees the secretary for other duties, it is counter-productive by

requiring additional documentation time from the physician and having the physician manually post-edit transcriptions is not cost-efficient.

In the post-editing scenario, a physician will dictate a diagnosis and transfer the recording to a server, like digital dictation described above. The recording can then be sent to an ASR service, either automatically or per the request of a medical secretary, and the secretary is presented a draft transcription to post-edit. While this approach does not handle as many challenges as real-time ASR, research from human and machine translation and translation dictation indicates that using draft output of either an ASR or machine translation system results in efficiency gains and reduces the time spent translating or transcribing a document.[2]

## 1.2 Contribution

The contribution of this thesis is a quantitative study of stød and an investigation of the technological application of stød. Specifically, the academic work addresses

1. Reliability of stød annotation (Chapter 3)
2. Ranking of acoustic features for stød detection (Chapter 4, Section 4.2)
3. Stød detection from acoustic input (Chapter 4, Section 4.3)
4. The technological application of stød in ASR by explicit modelling (Chapter 5)
5. Implicit modelling of stød using salient acoustic features (Chapter 6)

Statistical analysis of stød requires annotated data and the analysis is only feasible if the annotation is reliable. Inter-annotator agreement and annotator competence is analysed on a small phonetically-annotated data set, which includes stød annotation. Based on reliable annotation, several voice quality measures known to be predictive of acoustic events that can signal stød are analysed, and we identify 17 features which are predictive of stød in two data sets. Using different voice quality feature sets, stød detection is studied and the conclusion of the study is that stød detection is possible when formulated as a multi-class classification task. This formulation facilitates stød modelling in ASR systems where adding stød annotation to the entries in the phonetic dictionary improves large-vocabulary ASR performance. Finally, large-vocabulary

[2]See Zapata & Kirkedal (2015) for a description of translation dictation and similarities to medical dictation and the references therein, e.g. Martinez et al. (2014), for background on efficiency gains using ASR or machine translation and post-editing.

ASR performance on three data sets is further improved using acoustic features which were discovered to be salient for stød detection.

A baseline speech recogniser for Danish read-aloud speech was developed as part of the academic work conducted in this thesis. In an effort to stimulate ASR research and development for Danish language and in the interest of dissemination and reproducible research, the speech recogniser is made publicly available under a permissive license. The intention is to lower the access barrier for NLP-interested students and developers who wish to integrate ASR into products or services. Results reported in this thesis will also be more easily reproduced and ASR improvements documented and disseminated. Due to the lack of prior work and published results, we obtain state-of-the-art performance on each data set, but expect that commercially available ASR systems will be able to achieve better performance.

| Chapter | Purpose | Data set |
|---|---|---|
| Chapter 3 | Analysis | JHP sample |
| Chapter 4 | Feature selection | Parole48+DanPASS-mono |
| | 10-fold cross-validation | Parole48+DanPASS-mono$^{\ddagger}$ |
| | Evaluation | Parole48+DanPASS-mono$^{\ddagger}$, JHP sample |
| Chapter 5 | Flat start | train_120kshort |
| | ASR training | train$^{\dagger}$ |
| | Development/Tuning | Stasjon03$^{\dagger}$ |
| | Evaluation | Stasjon06$^{\dagger}$ |
| Chapter 6 | Flat start | train_120kshort |
| | ASR training | train$^{\dagger}$ |
| | Evaluation | Stasjon06$^{\dagger}$, Parole48, DanPASS-mono |

Table 1.1: The data sets used in the present work and their purpose. A data set will be introduced when it is first used, e.g. the JHP sample is introduced in Chapter 3. The symbol $^{\dagger}$ denotes disjunct subsets of the Språkbanken corpus. The symbol $^{\ddagger}$ denotes that these data sets are split into disjoint sets for different purposes in the same experiment. *train_120kshort* is a true subset of *train* (not disjunct).

A number of corpora are used in this thesis for different purposes and some are used both for training and testing. A summary of the data sets used and their purpose in a specific chapter is outlined below in Figure 1.1. We use corpora to evaluate annotation, train classifiers, analyse acoustic features, tune parameters and evaluate performance and the different purposes have varying requirements to annotation, corpus size and speech genre. For instance, speech recognisers need a lot of data to estimate good models and large speech

corpora often contain read-aloud speech, but the speech genres that we wish to recognise are dictation and spontaneous speech and we want to investigate if an improvement to a model can generalise to other speech genres because that gives us an indication that we are not overfitting to specific features of a data set.

## 1.3 Summary

Improved ASR systems for Danish can make it possible to use Danish voice control to interface with new technology such as wearables where keyboard or mouse interaction is not possible or appropriate. ASR can free resources and make medical dictation workflows more efficient while complying with relevant legislation. Whether using online ASR or offline ASR and post-editing, high accuracy Danish ASR is necessary to realise these efficiency gains and also important if Danish should continue to be a digital language. Stød detection can improve large-vocabulary ASR for Danish by distinguishing otherwise phonetically-identical lexical items. To assess whether stød can feasibly be detected from acoustic input, we first conduct a reliability study of stød annotation. This is followed by an investigation of the capabilities of acoustic features to predict stød as well as a series of experiments aimed at developing reliable stød detection. Lastly, we present a baseline speech recogniser and model stød in ASR models.

# Chapter 2

# Background

To understand stød and the models of stød which exist, Sections 2.1 and 2.1.1 will introduce phonetics, prosody and other terms that are necessary to understand the nature of stød. The theory and terminology is needed to understand the overview of previous stød-related research in Section 2.2 and technological applications in Section 2.3. Similarly, an introduction to acoustic terminology and a number of acoustic features is presented in Section 2.4. Because ASR is a complicated process and several theoretical and computational aspects are relevant to incorporate stød, this chapter will introduce ASR background, including acoustic, pronunciation and language models and decoders in Section 2.5.

### Occurence of stød

Stød is a remnant of a tonal system, which still exists in Swedish and Norwegian. Several features are in common between Swedish Tone-1 and Danish stød. An interesting dialectological fact is the absence of stød in some Danish dialects. Figure 2.1 shows the boundary between dialects where stød occurs and where it is absent.

On one side of the stød boundary, the information stød contributes to spoken communication is omitted. In place of stød, the semantics of a word can be resolved based on lexical context[1], e.g. articles or pronouns can be used as cues to the meaning of words as in *Jeg viser ham sølvtøjet* vs. *En viser på et ur* where *jeg* and *en* indicate the reading of *viser* (EN: I show him the silverware/A hand on a clock). We conjecture that this fact is the main reason stød has not been modelled utilised in technology. The distributional hypothesis is the basis for most statistical NLP, e.g. language modelling, information retrieval, search engines, statistical machine translation and many methods employed in ASR. However, there are cases where stød is the only inter-sentential cue to the reading of a sentence, e.g. stød is the only cue that distinguishes *de kendte folk*

[1]The distributional hypothesis: "You shall know a word by the company it keeps" (Palmer, 1968).

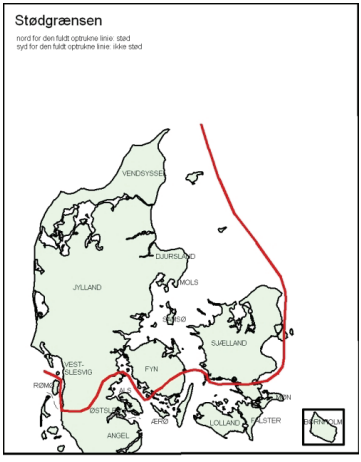



Figure 2.1: South and east of the red line, stød is absent from the regional dialects. Image from `http://dialekt.ku.dk/dialekter/dialekttraek/stoed/`, Dialectology Section, Department of Nordic Research, Copenhagen University.

(EN: The famous people) and *de kendte folk* (EN: They knew people)[2] in spoken Danish or *Ingen elsker bønner* (EN: No one likes beans) vs. *Ingen elsker bønder* (EN: No one likes farmers).

## 2.1 Phonetics

Phonetics is the study of human speech with physical sounds as focus. To be able to describe physical sounds to other linguists, a sound can be represented by a symbol. In phonetics, a sound is represented by a specific symbol regardless of the linguistic content. A phonetic alphabet makes it possible to describe and distinguish speech sounds.

Many *phonetic alphabets* have been proposed, but the International Phonetic Alphabet (IPA) (International Phonetic Association, 1999) is most prevalent. IPA was originally designed to be able to describe the sounds of all languages and is widely used in Danish phonetics, especially due to the fact that it is possible to describe Danish phonetics to phoneticians and linguists that are not Danish speakers. Easier communication via IPA also facilitates publication in academia.

IPA can be described using unicode encoding and also has two standardised mappings into ASCII encoding using either the (Extended) Speech Assement Methods Alphabet ((X-)SAMPA) (Wells, 1997) or

[2] An adverbial vs. verbal reading of *kendte*.

Kirshenbaum IPA (ASCII-IPA) (Kirshenbaum, 2001). Both mappings predate the widespread use of unicode encodings such as utf8 and utf16, but are still used in software programs and have been used to annotate many speech corpora. For instance. the open source speech synthesis program eSpeak (Duddington, 2012) uses ASCII-IPA and the multilingual EUROM1 corpus (Chan et al., 1995) is annotated in X-SAMPA. Table 2.1 illustrates some differences between the three alphabets. The alphabets share a large set of symbols that represent the same sounds, which can make it difficult to identify which IPA mapping is used from the alphabet itself, which is important because there is another subset of common symbols that do not symbolise the same same sound.

| Alphabet | Transcription |
|---|---|
| IPA | sɪmpəl |
| SAMPA | sImp@l |
| ASCII-IPA | sImp@L |

Table 2.1: Simplified phonetic transcription of the word *simple*.

Irrespective of the alphabet used, the sounds in an audio recording are represented as a sequence of symbols. This subbranch of phonetics is known as *segmental phonetics* and each symbol is called a *segment*. Segments are theoretically discrete and are ordered in time.

In this context, a *phone* is a segment. In the transcription in Table 2.1, each letter is both a phone a segment.

### 2.1.1 Prosody

A phone is the basic unit of speech. It describes vowels, consonants etc. A phone can be annotated with a diacritical symbol which represents a *suprasegmental* feature. Suprasegmental features can overlap segment boundaries (hence the name) and can overlap other suprasegmental features. Prominent examples of suprasegmental features are stress and stød. Suprasegmental features are often properties of syllables and are also known as *prosodic* features.

In Table 2.2, word stress is annotated as a suprasegmental feature using ['], ["] and [’], respectively. Each phone is separated by a whitespace to better give an indication of the differences between the mappings. Note that a segment annotated with prosody is also a phone, though the theoretical discreteness is compromised. A phone can be denoted as a complex phone to make suprasegmental annotation explicit.

While this could be an adequate description for a human linguist, it is problematic in computational terms. There is no annotation for the duration of a prosodic feature. Also, there is no notion of a syllable

| Alphabet | Transcription |
|---|---|
| IPA | d ˈɪ f ə k ə l t |
| SAMPA | d I" f @ k @ l t |
| ASCII-IPA | d 'I f I k @ L t |

Table 2.2: Phonetic transcription of the word *difficult*.

in segmental phonetics. However, prosodic features are often considered a property of a syllable rather than a phone. The linguist analysing a phonetic transcription must interpret on-the-fly. That phonetic resources are created for human consumption without accounting for computational uses is often a barrier for using phonetic resources in computer programs. An example of this is the difference between the annotation of word stress, which is affixed both to the left and to the right of the vowel which forms the core or nucleus of a syllable in Table 2.2. IPA annotation – or (X-)SAMPA annotation – does not imply a standard annotation scheme. The phonetic annotation of corpus $A$ may be different from corpus $B$ even though they use the same annotation alphabet. Mapping individual segments and suprasegments is inadequate and mapping between complex phones is necessary. If the affixation of suprasegments is unordered, several thousand complex phones need to be mapped. It is easy to spot the difference in the table, but discovering these annotation differences in large amounts of data is a difficult task.

The different annotation of prosodic features are repeated for Danish stød. Stød is annotated as [?] in SAMPA[3], [ʔ] in ASCII-IPA and [ˀ] in IPA. They are used as shown in Table 2.3.

| Alphabet | Transcription |
|---|---|
| IPA | tanˀ |
| SAMPA | tan? |
| ASCII-IPA | t?&n |

Table 2.3: Phonetic transcription of the Danish word *tand*.

The irregular affixation of suprasegmental features is a symptom of the fact that phonetic annotation has traditionally been created for human consumption and this is still the case in e.g. socio-linguistic studies. In some corpora, suprasegmental features may even be practically annotated as separate phones.[4]

[3]But stød is annotated as [!] in the DK-Parole corpus.

[4]E.g. in the corpus used in Chapter 3.

For computational purposes, where different sounds are represented by phones such as ASR or speech synthesis, the alignment between phone and sound is important. For some applications such as ASR, an alignment can be induced using embedded training or forced alignment. The specifics of these methods are explained in Chapter 5.

### 2.1.2 Acoustics

The subject of interest in acoust ics is sound waves. An oscillation is one cycle of repetitive variation in time of a sound wave. If an acoustic signal is e.g. a musical note, there will be many oscillations per second. The musical note A has a *frequency* of 440 Hz because the sound wave oscillates 440 times per second. Frequency is the acoustic correlate of *pitch*. The *pitch period* is the duration of one oscillation. If no pitch can be detected, there is little or no repetitive oscillation meaning the sound wave is not periodic.

*Amplitude* is a measure of the change in atmospheric pressure that is caused by sound waves. Amplitude is the difference from the peak of an oscillation to the 'centre' of a sound wave. The mean amplitude over a time window is called *intensity*.

The human voice produces complex signals. A periodic signal created by a human has a fundamental frequency and component frequencies. The component frequencies are integer multiples of the fundamental frequency and called a *harmonic*. If the fundamental frequency is 100 Hz, the 2nd harmonic is at 200 Hz, the 3rd harmonic at 300 Hz etc.

The air flow through the glottis is called the *glottal flow*. Glottal analysis is a method to estimate glottal flow parameters that characterise the voice source. Feature that describe voice quality can be extracted from the voice source.

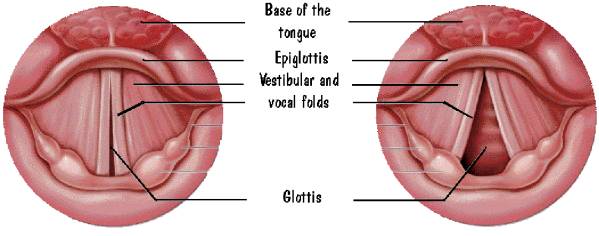



Figure 2.2: Illustration of the larynx where we can see the glottis as the opening between the vocal folds. Image from `http://roble.pntic.mec.es/~mfec0041/bachillerato/archivos/web_phntks/files/theory_notes.htm`.

To analyse a complex acoustic signal, the signal must be decomposed. Several decomposition methods exist. In ASR, Discrete Fourier Transformation is used, but other disciplines such as speech synthesis may use the Continuous Wavelet Transform. In essence, the two methods decompose a complex signal into component signals at different frequencies.

## 2.2 Acoustic investigations into stød

As mentioned in Chapter 1, stød is a perceptual feature. In a similar fashion, pitch is a perceptual feature whose primary acoustic correlate is fundamental frequency (F0). Linguistic studies of stød, where the acoustic realisation of stød has been the focus of the investigation, have found no single acoustic correlate for stød (Hansen, 2015).

The most robust correlate is an abrupt decrease in intensity that is related to constriction of the glottal flow. Another strong indicator is irregular vibration of the vocal folds which produces a creaky sound. However, the correlation is not perfect, as stød can be perceived without the presence of creak or irregular vibration (Hansen, 2015). Stød can also be audibly perceived yet not be visible in a waveform signal (Riber Petersen, 1973; Grønnum & Basbøll, 2007)

The absence of a single correlate has given rise to a substantial literature on the subject and the description of stød is an active research area. Below is an outline of two descriptions of stød.

### 2.2.1 Stød description

The current description of stød stems from investigations in Fischer-Jørgensen (1989). Stød-bearing syllables are divided into two phases with stød manifestation on the second phase. In the case of a long vowel with stød, the two phases divide the long vowel in two temporally equal parts. In the case of a short vowel and a sonorant consonant[5], the boundary between the two phases coincide with the segment boundary. These two alternative prerequisites – a long vowel *or* a short vowel and a subsequent sonorant – are collectively known as *stødbasis*. If none of the prerequisites are present, stød cannot be manifested.

Danish stød has been studied in a series of publications by Nina Grønnum (Grønnum & Basbøll, 2001, 2002, 2003; Grønnum, 2006; Grønnum & Basbøll, 2007, 2012; Grønnum et al., 2013) together with Hans Basbøll. The research is based on the concept of stødbasis and included in Grønnum (2005) which is a Danish textbook on phonetics and phonology.

[5] A sonorant consonant is either a nasal, lateral or r-sound, e.g. [m], [l] or [ɾ] (Grønnum, 2005). In practice, phonetic annotators sometimes relax this constraint to any sonorant.

A different account of stød stems from Hansen (2015). His description is based on Ladefoged's phonation types.

The stødbasis and phonation-based accounts of stød are outlined below.

#### 2.2.1.1 Ballistic model

A syllable has the potential for stød if it has stødbasis. There can only be one stød per syllable and polysyllabic words can have more than one stød. Grønnum (2005) describes the realisation of stød in two acoustic events:

- glottal stop
- creaky voice

**Glottal stop** is an instant of glottal closure where the vocal folds are closed and prevent airflow through the vocal tract. In colloquial English, a glottal stop can replace a [t] in words like *mountain* or *metal*. In Danish, it can also signify the realisation of stød in extreme cases according to Grønnum & Basbøll (2007).

**Creaky voice** describes a type of phonation. The vocal folds of a human can be open and allow for maximum airflow or be closed and prevent airflow. In either state, the vocal folds do not vibrate. In between maximum and zero airflow are degrees of openness that determine the vibration of the vocal folds when uttering sonorants. When the vocal folds constrict airflow and are relaxed, vocal fold vibration is not completely harmonic. The slight disharmony on an otherwise harmonic acoustic signal sounds like a 'creak' and gives rise to the name *creaky voice*.

Grønnum & Basbøll (2007) describe stød phonetically as a ballistic gesture which minimally generates a slightly compressed voice or maximally a distinct creaky voice and a glottal stop under emphasis, as aligned with syllable onset and a property of the syllable rhyme. The ballistic gesture is a muscular response to a neural command that, once executed, the speaker can no longer control.

#### 2.2.1.2 Phonation-based model

A way to describe voice quality is to use phonation types. There is a continuum of the degree of openness of the vocal folds that span from closed as is the case with the glottal stop, and most open, where the vocal folds do not vibrate and airflow passes unhindered. The degrees of openness of the vocal folds are binned into different phonation types:

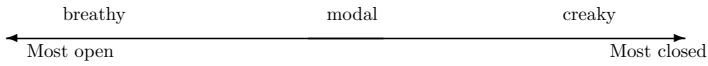



Figure 2.3: Voice quality scale after Gordon & Ladefoged (2001).

**Breathy voice** is a type of phonation where the vocal folds are far apart, do not constrict airflow and vibrate very little.

**Modal voice** is often described as the optimal degree of openness and vibration for sonorants.

This model describes stød as a correlation with voice quality at the scale between modal and creaky voice. Realisation of stød as a glottal stop would be one extreme and modal voice would be the absence of stød and the other extreme. Hansen formulates his hypothesis as:

> "The hypothesis is that stød is expressed as a relatively short change in voice quality towards a more compressed e.g. creaky voice quality and subsequently returns to less pressed voice quality. Hence, stød is treated as a dynamic voice quality gesture. A well-formed stød is a suitably large fluctuation in voice quality over a suitably (short) time frame. Whether creak occurs in connection with stød or not depends on where on the voice quality scale the stød fluctuation starts."[6]

*Creak* is a term for a phonation type between creaky and modal voice, but also denotes a suprasegmental feature. Unfortunately, Hansen must reject his theory after rigorous evaluation and the phonetic description of stød remains unclear.

#### 2.2.1.3 Ballistic vs. phonation model

The ballistic model describes how stød is produced and how stød manifestation can vary according to the strength of a neural command. The phonation-based model is a dynamic voice quality gesture that accounts for the manifestation of stød and explains why stød manifestations which are acoustically dissimilar are perceived similarly. So the ballistic model accounts for articulation and production and the phonation-based model also accounts for perception, but the two models are only mutually exclusive in the production account, i.e. a ballistic gesture vs. a voice quality gesture.

[6]The author's translation of the hypothesis in Hansen (2015) from Danish to English.

The two explanations or descriptions are relevant to this study because we will be using data sets annotated with stød that has been created manually and mainly annotated by students of Nina Grønnum. We assume they have applied, or at least been influenced by, her theories. This may be beneficial if annotators use the same method to annotate, but the annotation conventions that a theory or method applies may be a source of error. As described above, stød manifestation coincides with the segment boundary if stødbasis is a short vowel followed by a sonorant consonant. The convention is to annotate stød on the sonorant consonant, but if the stød manifestation is not prototypical and stød is manifested on the vowel, we do not know if the annotator follows convention or his/her aural perception.

While Hansen uses a very small data set from a single speaker, his work is the most thorough acoustic/-phonetic research available. Hansen seeks a characterisation of stød and though he rejects his hypothesis, his observations has guided the methodology chosen in Chapter 4.

## 2.3 Stød-related technological applications

There have been no major uses of stød in technological applications except in speech synthesis. It seems reasonable to attribute this to the variable manifestation of stød. However, glottal information that indicates the presence of creak in speech has been explored in ASR previously (Yoon et al., 2006; Riedhammer et al., 2013). Detecting or exploiting creak in ASR is therefore the most similar technological application, because it is one of the acoustic events used to describe stød.

Yoon et al. (2006) used the measure H1-H2 and mean autocorrelation ratio $r_x$ in a decision algorithm for voice quality. The decision algorithm assigns one of three labels to 10 ms samples extracted from the Switchboard corpus (Godfrey et al., 1992):

**Voiceless** All samples where no pitch could be detected

**Creaky** Samples where H1-H2 $< -15$ dB or (H1-H2 $< 0$ db and $r_x < 0.7$)

**Modal** All other samples

Including voice quality in ASR experiments improved word recognition accuracy for American English. Yoon and colleagues also investigated whether Perceptual Linear Prediction (PLP) coefficients are salient for classifying creaky vs. non-creaky sonorants using a support vector machine classifier with a radial basis function kernel. Many classifiers make use of a distance function to compute similarity between two samples $x$ and $x'$, i.e. a small distance measure signifies greater similarity. Kernels compute a similarity measure with an upper bound of 1 where $x = x'$ and zero. The similarity measure of an radial basis function kernel is calculated as

$$K(x, x') = \exp(\gamma||x - x'||^2) \tag{2.1}$$

$||x - x'||$ is the euclidean distance between two vectors and $\gamma$ is a free parameter that can be tuned using grid search on a development set. Using no parameter tuning and 1v1 evaluation, the experiment showed that PLP features alone contained information to distinguish between creaky and non-creaky versions of sonorants.

Creakiness in American English does not signal lexical contrast as stød does, but is a marker for lexical, syntactic and prosodic boundaries (Redi & Shattuck-Hufnagel, 2001). It does indicate that information about glottalisation can inform ASR and, if stød can be predicted or detected with confidence, it could improve Danish ASR because distinction can be made between stød-bearing and stød-less variants. As a function of the syllable, the realisation of stød crosses segment boundaries but can also in extreme cases cross word boundaries in colloquial speech due to elision of word endings and contraction of adjacent words into a single phonetic word. A common example from Danish is the phrase *der er* which is merged to a single phonetic word [dɑːˀɐ].

For the Austronesian language Tagalog, an investigation into recognition of the glottalisation phone [ʔ] was conducted (Riedhammer et al., 2013). The study showed that a 1-state model rather than the linear 3-state model[7] was appropriate for modelling [ʔ] because the duration of [ʔ] is 10-40 ms and frequently shorter than the minimum duration enforced by the 3-state models topology[8]. The study also showed that deleting [ʔ] or merging it with the subsequent phone led to an increase in word error rate (WER) and an artificially large phone inventory.

The short glottalisation or creak in Tagalog is not consistent with Danish stød and because Danish stød tends to cross segment boundaries and have a longer duration, a 1-state model is not a logical choice to model stød.

## 2.4 Acoustics

From previous studies outlined above, the voice quality measures F0, intensity and H1-H2 have been shown to correlate with stød to some degree. However, we do not know what stød is and therefore it is difficult to choose an acoustic feature that describes it. We therefore intend to study a number of features in Chapter 4. These features are introduced in this section.

[7] See Chapter 5 for an explanation of the 3-state model.

[8] At least one 10 ms sample per state is necessary and $3 * 10ms = 30ms$.

To analyse a speech signal and extract acoustic features, short-term acoustic analysis is used to decompose a continuous signal in time for computational processing. Recordings are divided into samples at regular intervals. The regular interval is known as the sampling *shift* and in this thesis, 10 ms is chosen because it is the standard shift used in ASR. The sampling is illustrated in Figure 2.4.

The time window in the illustration is 25 ms i.e., there is a substantial information overlap in the features calculated. The window is larger than the shift because feature estimation algorithms sum or integrate over the time window to estimate, e.g. energy, pitch or MFCC features. The window size is a trade-off because a large window makes features more robust but too large a window makes the computation less sensitive to small variations. Different acoustic measures also require different window sizes as explained for F0, phase features and harmonics-to-noise ratio in Section 2.4 below.

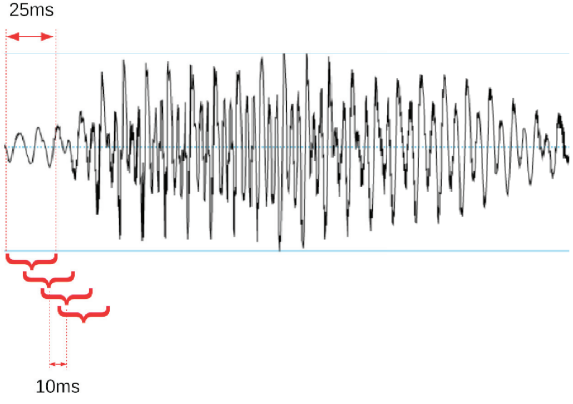



Figure 2.4: An illustration of the sampling frequency and the time window (25 ms). Sampling frequency (10ms) is also known as the sampling shift.

### 2.4.1 Voice Quality features

The features investigated in Chapter 4 will be described in this section. First, two basic acoustic features used in previous studies are outlined and followed by several features used to describe voice quality. Two features extracted from the phase spectrum in the speech signal are described next and followed by a short description of standard ASR features.

**Fundamental frequency**

Pitch tracking is a non-trivial task. It is based on fundamental frequency (F0) which is the primary acoustic correlate of pitch. Harmonics above F0 also have an impact on the perception of pitch but in practical terms, pitch tracking or pitch detection is equivalent to F0 estimation (Gerhard, 2003).

F0 is the frequency of the vibration of the vocal folds. To find F0 and other harmonic frequencies, the speech signal must be decomposed into frequency components. A short-time Fourier analysis (Allen & Rabiner, 1977) or adaptive Harmonic Model (Degottex & Stylianou, 2013) can decompose a complex sound wave into the component waves that compose the original signal.

F0 estimation requires a time window that is longer than 25 ms to extract robust features. For modal phonation, speakers can generally be expected to produce F0 values above 62.5 Hz which can be captured by a 25 ms window. In creaky phonation, F0 values can be as low as 10 Hz and that requires a longer time window to capture at least 2 pitch periods (Kane & Gobl, 2011).

**Harmonics-to-Noise ratio**

Harmonics-to-noise ratio is used to estimate the level of noise in human voice signals. Harmonics-to-noise ratio is the degree of periodicity in speech vs. the amount of noise on a logarithmic scale and is calculated over six pitch periods (Boersma, 1993). The time window that is considered for the calculation of the harmonics-to-noise ratio is also larger than 25 ms. Hansen (2015) uses the harmonics-to-noise ratio as a confidence measure for F0 estimation and also as an estimate of irregular vibration in the vocal folds which frequently occur in connection with stød.

**H1-H2**

H1-H2 is the difference between the amplitudes of the first two harmonics. H1-H2 is a spectral cue that characterises creaky phonation when the amplitude of the second harmonic is higher than the amplitude of the first harmonic (Yoon et al., 2006), i.e. when the differnce is negative. The first harmonic is practically implemented as the harmonic peak closest to the estimated F0 and the estimation of H1-H2 therefore relies heavily on F0 estimation. Note that there is a related measure - H1:H2 - which is a ratio between the first and second harmonic and that both H1-H2 and H1:H2 is sometimes denoted H1H2 in the literature.

**Quasi-Open Quotient**

Quasi-Open Quotient describes the relative open time of the vocal folds. Quasi-open quotient is the duration where the glottal flow is at least 50% above the minimum flow and normalised by the pitch period.

**Normalised Amplitude Quotient**

Normalised Amplitude Quotient describes the glottal closing phase. It is a ratio between the maximal glottal flow and the minimum of its derivative normalised by F0. It is a robust and efficient parameter to separate phonation types as reported in Drugman & Dutoit (2010). $R_d$ (See below) is described as "quasi-similar" to the normalised amplitude quotient.

**$R_d$**

The basic shape parameter $R_d$ is qualified as "the most effective parameter to describe voice qualities in a single value" (Fant, 1995). A low $R_d$ value is related to effective glottal closure and high $R_d$ is associated with abducted phonation, e.g. voiceless phones. The complete description of the parameter is beyond the scope of the thesis and the reader is referred to the paper for an in-depth description.

**Maxima Dispersion Quotient**

Maxima Dispersion Quotient is a parameter designed to quantify the dispersion of the Maxima derived from the wavelet decomposition of the glottal flow in relation to the glottal closure instants.

**Parabolic Spectral Parameter**

Parabolic Spectral Parameter quantifies the spectral decay of a glottal pulse in the frequency domain with a parabolic function. The spectral decay of a glottal pulse is normalised with respect to a hypothetical maximal spectral decay of the direct current flow[9] of the same signal, which is dependant on F0. By normalising, the parabolic spectral parameter can be used to compare glottal sources with respect to spectral decay, even though the voices have different F0 and has been shown to correlate with phonation types (Fernandez, 2003).

**Peak Slope**

After applying an octave filter bank with filters centered at 8 kHz, 4 kHz, 2 kHz, etc. until 250 Hz, the local amplitude maximum for each band is computed. Peak Slope is the slope of a straight regression line fitted to the peaks of the speech segment. The slope of the regression line will differ depending on whether the phonation type is breathy, modal, tense etc. In comparison, if the amplitude peaks were only H1 and H2, the measure should be similar to H1-H2. H1-H2 computation depends on F0 estimation, which is not the case for Peak Slope. Hence, Peak Slope should be better suited to non-modal speech segments (Kane & Gobl, 2011).

[9]Direct current flow is the airflow before modulation by the glottis.

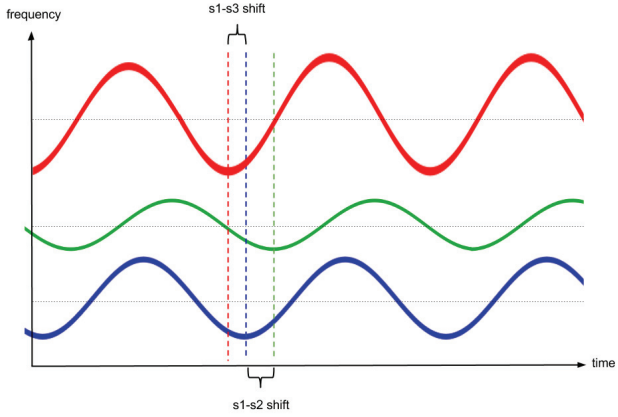



Figure 2.5: Illustrates waves that are not in phase. The difference between s1 and s2 or s1 and s3 is called phase shift, analogous to time shift in Figure 2.4.

### 2.4.2 Phase features

Phase information places a sound wave in time. The frequency components of a naturally occurring complex sound wave are not completely in phase. Phase distortion is derived from the computation of relative phase shift, which is the desynchronisation between the first harmonic $h_1$ and harmonics at higher frequencies $h_n$. The phase distortion at instant $i$ is calculated as

$$PD_{i,h} = \phi_{i,h+1} - \phi_{i,h} - \phi_{i,1} \tag{2.2}$$

where $\phi_{i,h}$ is the instantaneous phase of harmonic $h$.[10] The phase distortion or phase shift is illustrated in Figure 2.5.

In the source-filter model of speech[11], phase distortion represents the shape of the source. While this is similar to the features in Section 2.4.1, phase distortion is independent of F0 and insensitive to the position of the glottal pulse and hence the position of the analysis window (Degottex & Erro, 2014).

To create robust parameters from phase distortion in short-term acoustic analysis, PDM and PDD are suggested in Degottex & Erro (2014) (see below). The data is assumed to be circular and obey a wrapped normal distribution for the calculation of mean and variance:

[10] In practice, $h$ is instead $K$ frequency bins – similar to estimation of Peak Slope.
[11] See e.g. the textbook Jurafsky & Martin (2008), Section 7.4.6.

#### PDM

Phase Distortion Mean must be estimated over a number of adjacent frames to be robust. Degottex & Erro (2014) uses a 25 frame window to ensure six periods are covered in the computation. Several frames are necessary to calculate mean values and a large context decreases sensitivity to noise in unvoiced segments and separates smooth behaviour from randomness of the phase.

#### PDD

Phase Distortion Deviation is estimated over a shorter context than PDM. PDD is intended to model the noise of the voice source and a wide window can cover the beginning of a voiced segment in addition to be sensitive to a longer trend in the speech signal which is modelled by PDM. This would result in an overestimated PDD. PDD is therefore estimated over 9 frames (roughly 2 periods) and the trend – modelled as PDM over the same window – is subtracted from PD before estimating PDD.

### 2.4.3 Automatic speech recognition features

Standard ASR features include Mel-Feature Cepstral Coefficients (MFCC) and Perceptual Linear Prediction (PLP) coefficients (Hermansky, 1990). Refer to Section 2.5 below for a description of the feature extraction and a description of these features.

#### Probability-of-voicing

Probability-of-voicing is a feature that assigns a probability to each frame that indicates whether the frame is voiced. Technically, probability-of-voicing is warped Normal Cross-Correlation Feature. Probability-of-voicing is also a glottal flow parameter, but mentioned here because it is extracted using an ASR system rather than software dedicated to speech analysis.

#### Pitch

The pitch feature is log-pitch with probability-of-voicing-weighted mean subtraction over a 1.5 s window.

#### ΔPitch

ΔPitch is the first derivative of the log-pitch without mean subtraction.

## 2.5 Automatic speech recognition

Speech recognition has been an active research area for decades. Many statistical methods have been applied to natural language for the first time in ASR and later found application in written language. Hidden Markov Models (HMM), Expectation-Maximisation algorithm and alignment/translation models were introduced in ASR as early as the 80s (Rabiner & Juang, 1986). The same methods later found application in machine translation in the VerbMobil project and shifted the focus in NLP from rule-based machine translation to statistical machine translation. Currently, the methods are used in many applications of NLP.

ASR systems perform a complex multi-step translation from sound waves to text that requires knowledge of signal processing, phonetics, statistics and computer science. Based on the approach used in the Kaldi toolkit, this background chapter will explain feature extraction in the *frontend* component, the acoustic model (AM), pronunciation modelling with phonetic dictionaries and phonetic decision trees (PDT), and syntax in language models (LM).

Figure 2.6 is a flowchart visualisation of how an ASR system is built. The necessary input resources is are speech, parallel transcripts, a phonetic dictionary, additional text and a so-called HMM topology (the green boxes in Figure 2.6). Red boxes are intermediate products and models that we convert to a Weighted Finite State Transducer (WFST) representation that are visualised as blue circles. The arrows show what resources are transformed or used to estimate a model or WFST. Arrows between WFSTs represent a mapping known as finite state composition and the process and the WFST names will be explained in Section 2.5.5.

If a box or circle has a border, the model of WFST is used in the ASR system.

#### Terminology

The terms used in speech technology and phonetics are similar yet different. While phonetic experts can distinguish between phones, phonemes and prosody, the distinction has been blurry in academic literature on speech technology. A historical lack of communication between the two research fields has given rise to parallel terminology. This chapter will use ASR terminology, but attempt to give a translation to phonetics if possible, either in the body of the text or in footnotes.

In ASR, the terms *phones* and *phonemes* have been used interchangeably. Arguments for using one or the other can be made, but due to the inherent close relation to sound and the realisation of spoken language, phones or phonetic symbols will be used to denote the symbols that make up a phonetic transcription, also denoted a *phonetic representation* or a phone sequence.

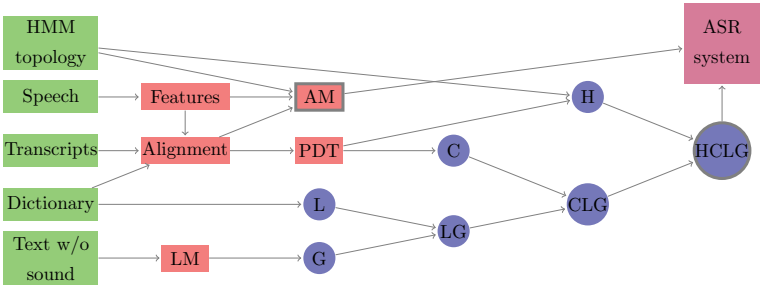



Figure 2.6: A flowchart that describes the necessary input resources, intermediate models and WFSTs created when training a WFST-based ASR system. The green rectangles are input resources, the red triangles are intermediate data representations and models estimated using the input resources and the blue circles are WFSTs. Triangles and circles with a black border are used in the ASR system.

ASR systems are distinguished with respect to speakers into 3 categories: speaker-dependent, speaker-independent and speaker-adapted. This categorisation was introduced in Woodland (2001) but different definitions are used in the literature.

Speaker-dependent systems are trained using data from a single speaker for use by that specific speaker. This type of system is not common because the accumulation of sufficient training data for the particular speaker is costly and time-consuming. Speaker-dependent systems are still in use in respeaking scenarios in the media industry. Accessibility requirements in public media such as TV requires subtitling of many programmes. Re-speaking exploits a time delay between the recording (or playback) of a programme and the actual broadcasting to viewers. In the delay, the speech in the programme is spoken aloud by a respeaker and speaker-dependent ASR specifically tailored to the respeaker create subtitles. Speaker-dependent systems usually achieve a high accuracy for the specific user but generalises poorly to other users.

Speaker-independent systems generally cannot achieve the same accuracy as speaker-dependent systems. However, accumulating data from many speakers across age, gender, dialect, etc. is considerably easier than for a single speaker and the resulting ASR system can be used by more than a single user and generally perform better on users not in the training data.

Speaker-adapted systems are speaker-independent systems that are adapted to a specific user using only a small amount of speaker-specific data. Practically, the user reads aloud a fixed set of sentences to create a development set. This set is used to tune model parameters to maximise recognition for the users voice.

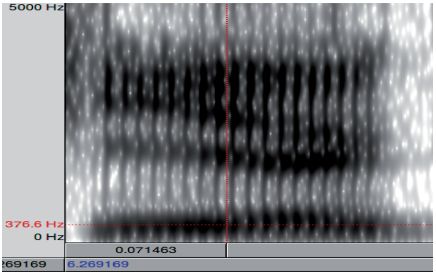



Figure 2.7: Spectrogram for the Danish word *cirkel* (EN: circle). Time is on the X-axis and frequency on the Y-axis. The spectrogram for vowel [i] can be seen left of the vertical dashed red line.

Speaker-adaptation will be relevant for stød. Supposing there are different strategies to realise stød, the strategies may be specific to region, speaker or social demographics.

### 2.5.1 Feature extraction

Feature extraction is exemplified by Mel-feature cepstral coefficients (MFCC). The sampling is the same short-term acoustic analysis illustrated in Figure 2.4.

For each sample, a frequency spectrum is computed by a discrete Fourier transform. The black colorations are called *formants* and the darkness indicate energy peaks in particular frequency bands. Figure 2.7 shows the spectrum of the vowel [i]. The lowest formant is F0.

A vertical cross section of the spectrogram shows a spectral profile for a given sound, see Figures 2.8 and 2.9. The energy distribution describes the sound uttered in a sample and can be used to classify samples as phones. Mel filters extract log-energy coefficients computed over specified intervals according to the Mel scale, which is linear below 1000Hz and exponential above. This models human hearing which is less sensitive to changes in high frequency bands. A Mel-filterbank consists of 20-40 filters.

After passing through Mel filters, each coefficient is converted to a cepstrum using an inverse Fourier transform or Discrete Cosine transform. The Discrete Cosine transform separates the contributions of the source and the filter and the coefficients in the cepstrum describe the filter, i.e. the vocal tract.[12] The MFCC features are the first 12 coefficients – not including the 0th coefficient – and an energy coefficient.

[12]The features in Section 2.4.1 describe the source.

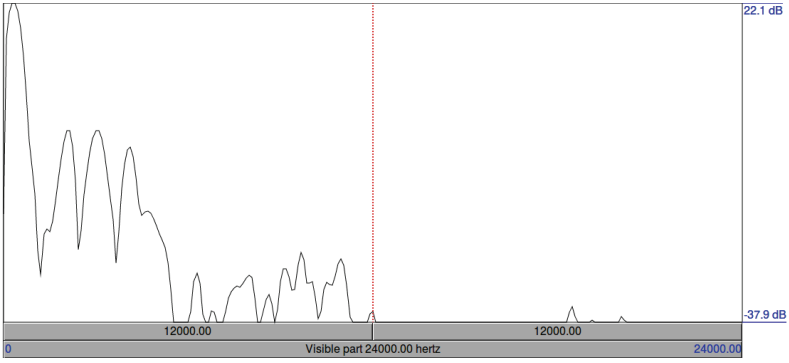



Figure 2.8: Spectral profile of the vowel [i].

#### 2.5.1.1 Feature transformation

Feature transformation is not explicitly stated in Figure 2.6, but represented by the arrow from *Features* to *AM*. Theoretically, one sample is assumed to be independent of the next, which is an assumption that is known to be false, but works well in practice. However, feature transformation of MFCC vectors can account for context-sensitivity.

MFCC features are sensitive to coarticulation, which is the effect of the temporal left and right phonetic context on the realisation of a speech sound. The pronunciation of [m] in *similar* is different from [m] in *summary*. The configuration of the speech organs in the filter (vocal tract) when uttering [i] in *similar* colours the pronunciation of [m] because the organs must move from one configuration to the next. The lips, tongue, teeth etc. never achieves the ultimate placement for the pronunciation of [m] before the utterance is finished, because 1) the place speech organs are moving from, and 2) the speech organs are anticipating the next configuration for uttering [i] and already moving towards that configuration.

$\Delta$MFCC and $\Delta\Delta$MFCC derivatives are computed specifically to model the effect of context and coarticulation or specifically, the speed and acceleration of speech organs such as tongue, lips, teeth, etc. The $\Delta$MFCC and $\Delta\Delta$MFCC derivatives are usually computed over a sliding window and appended to the MFCC features to create a 39-dimensional vector.

To model longer-range context than $\Delta$+$\Delta\Delta$ features, we can apply Linear Discriminant Analysis to a spliced feature vector. Feature splicing concatenates feature vectors over a time window specified in samples, e.g. +/- 3 samples which becomes a time window of 70 ms. $((3+1+3)\times 10\text{ms} = 70\text{ms})$. Linear discriminant

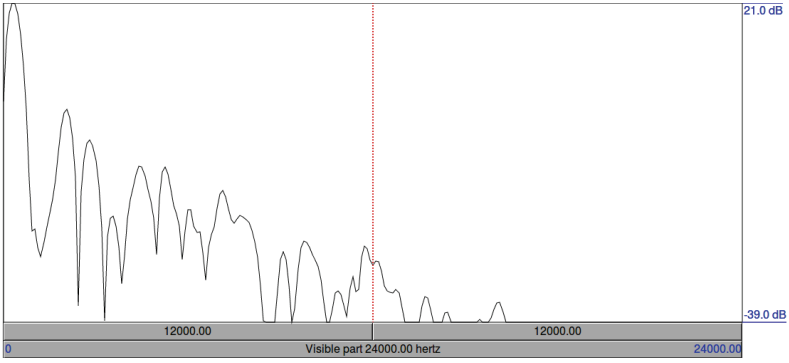



Figure 2.9: Spectral profile of the vowel [e].

analysis can perform dimensionality reduction to a new feature space and the features from this space is then used insted of MFCC vectors.

### 2.5.2 Acoustic modelling

Recently, artificial neural network (NN) AMs have begun outperforming Gaussian Mixture model (GMM) AMs (Hinton et al., 2012), but GMMs are still used to create the initial alignment from vectors to phones and this description of acoustic modelling will be based on GMM+HMM AMs.

Creating an ASR system requires an alignment between speech and orthographic transcriptions as illustrated in Figure 2.6. In a first step, word sequences in the transcripts are mapped to phone sequences using a phonetic dictionary. The dictionary must cover all the words in the training transcripts or have access to a grapheme-to-phoneme converter. Phone sequences are aligned to feature vector sequences by assuming an equidistant segmentation of feature vectors based on the sequence of phonetic symbols. This is known as *flat start*. Based on this alignment, an AM is estimated. In Kaldi, Viterbi forced alignment[13] assigns each feature vector to a phone. After alignment, feature vectors are clustered by phones and an AM is estimated. Using the AM, a new alignment is computed by re-classifying samples, which are then clustered, and this clustering is used to estimate another AM. This iterative refinement can be repeated for a fixed number of iterations or until convergence, i.e. the alignment does not change. At specified non-consecutive intervals, the training data is resegmented based on the current AM.

[13]Baum-Welch estimation using the backward-forward algorithm can also been used, but Viterbi is less computation-intensive and produces comparable results with sufficient data.

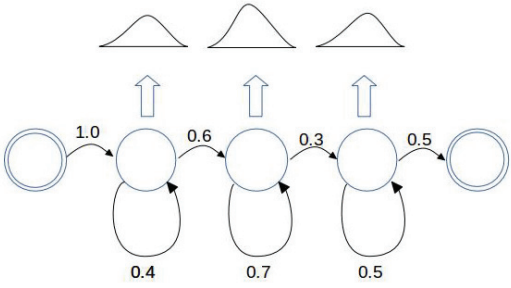



Figure 2.10: HMM model for [l].

#### 2.5.2.1 HMM states

Phones are modelled using HMMs and the HMM topology is necessary to estimate the AM. They are the building blocks of the AM and the standard topology is illustrated in Figure 2.10. The figure is a 5-state model, where the initial state and final state are illustrated using two circles. The three middle states are denoted as $B$, $I$ and $E$ (For Begin, Internal and End) and are emitting states, where the initial and final state are non-emitting. This 5-state topology is similar to a 3-state topology with only emitting states, but we use a 5-state topology for illustrative purposes. The parameters of an HMM are transition probabilities and output distributions (for emitting states). The HMM will output a continuous vector (if we are in an emitting state) and randomly change state to the next state or the same state according to transition probabilities at each time step.

Phones are trivially projected to HMM states before aligning feature vectors to phones sequences as described above. The individual phone HMMs are concatenated to create a state sequence corresponding to sound input. This alignment is used to estimate emission probabilities and transition probabilities. The transition probabilities on arcs leaving a state should sum to 1 and emission probabilities are modelled with a Gaussian or a GMM.

An AM may model a phone with several HMMs – one for each context a phone is observed in, i.e. one or two phones to either side.[14]

The choice of HMM topology enforces a minimum duration on a phone. If the sampling interval is 10 ms, the minimum duration is 30 ms for HMMs with 3 emitting states and 50 ms for HMMs with 5 emitting

[14]See Section 2.5.3.1 for further explanation of phonetic context dependency.

states. In Kaldi, phonetic symbols that are intended to model silence, background noise and other sounds not related to speech are modelled using 5-state HMMs and non-silence phones use a 3-state topology.

The self-loops on $B$, $I$ and $E$ model the n-to-1 relation between samples and HMM states. It is not possible a priori to know the duration of a phone and it will often be more than 3 samples. Implicit duration modelling could be achieved by only adding a loop on $I$, but there is considerable variability between $B$, $I$ and $E$ in plosives for instance, and if it is not possible to transition back in $B$ and $E$ states, the $I$ state must model substantial variability. Like the duration of $L$ cannot be determined, the duration of each state cannot be determined and therefore it must be possible to transition from $B$ to $B$ and from $E$ to $E$.

### 2.5.3 Phonetic dictionaries

The phonetic dictionary is a mapping between words and their phonetic representation(s) and the set of classes a vector can be classified as are represented in the phonetic transcriptions. The phonetic dictionary is also known as the lexicon and sometimes the pronunciation model (PM), but may also entail phonetic decision trees. In this thesis, we use the latter definition of PM, i.e. it is both the phonetic dictionary and the phonetic decision tree because the phonetic decision tree models phonetic context.

#### 2.5.3.1 Phonetic context

The phonetic symbols in the phonetic dictionary are context-independent and can be mapped to a context-dependent (CD) phonetic representation. A monophone can be divided according to context. To model long-range influence from context, monophones can be divided into *word-position-dependent phones*. Word-initial and word-medial pronunciation of a phone is different than the word-final pronunciation (Liu et al., 2011). A monophone [$e$] would be divided into word-initial [$e_B$], word-internal [$e_I$], word-final [$e_E$] and singleton [$e_S$] phones.

Local CD clustering is known as *triphone* modelling. A monophone [i] in the word *similar* is divided into two triphones: $s{-}i{+}m$ ([i] preceded by [s] and followed by [m]) and $m{-}i{+}l$ ([i] preceded by [m] followed by [l]). This subdivision is carried out for all combinations of phones to create a set of *triphones* which is a phone in a specific local phonetic context. The left and right context window can be increased and a *pentaphone* uses two phone-context to cluster training data.

For word-position-dependent phones, if the sequence [$l_I$ $a_I$ $i_E$] occur in the training data, $a_I$ is mapped to the triphone $l_I{-}a_I{+}i_E$.

The triphone subdivision reduces the variability of the data and position-dependent phones can make classification less robust if faced with data sparsity or make the number of classes explode and make

classification intractable. To counter this problem, the states in triphone HMMs can share Gaussians across triphones via *state-tying*.

**State-tying**

The phonetic difference between states in the triphones $s_B{-}i_I{+}m_I$ and $s_B{-}i_I{+}n_I$ might not be relevant for recognition purposes and merging HMM states reduces the number of parameters that need to be estimated during AM training. After merging, the phone models are called *tied-state* triphones.

State-tying is performed in 4 steps that ties into the AM and uses the phonetic dictionary (Young et al., 1994):

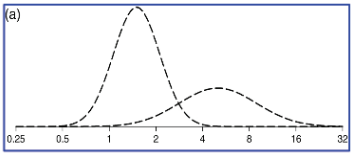
1. Train a monophone AM where each HMM state is modelled by a single Gaussian.
2. Copy each monophone HMM for each phonetic context and re-estimate triphone models using the trained monophone parameters for initialisation.
3. Cluster corresponding triphone states derived from the same monophone HMM based on *phonetic similarity* into tied-states and choose one HMM state as prototype.
4. Increment the number of GMM components in each tied-state and retrain until a fixed number of mixture components is reached or the log-likelihood improvement is below a threshold.

State-tying relies on merging phonetically similar states (step 3), but the notion of phonetic similarity has yet to be defined. In the first step, the word-position-dependent, stress-dependent, tone-dependent triphones and other CD variants of the same monophones or *base* phones are clustered in the root node of a *phonetic decision tree* (PDT). The data represented by the phones in the root node is split by asking yes/no questions about the phonetic context and grown into a binary PDT. The question that maximises the log-likelihood of the data is chosen as splitting criterion, i.e. a greedy top-down algorithm chooses the locally most optimal question under the assumption that the data in each child node is modelled by a single Gaussian.

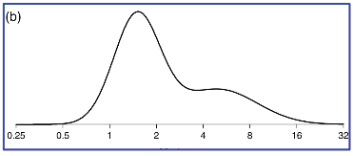
The PDTs are grown until the likelihood increase falls under a threshold $S(o)$ or a minimum state occupancy threshold is reached. It must be possible to accumulate enough statistics for all states in the node after node splitting and hence the minimum state occupancy threshold is imposed.

In the final step, the decrease in likelihood by merging leaf nodes is calculated and all pairs of nodes for which the decrease is below $S(o)$ are merged. Subsequently, step 4 of the state-tying process can begin.

The questions that define phonetic similarity can be questions about the right and left symbolic phonetic context. The left-most HMM state in the triphones mentioned above, $s_B{-}i_I{+}m_I$ and $s_B{-}i_I{+}n_I$, are




Two Gaussian distributions with different means and variances.




A non-normal distribution.

Figure 2.11: Non-normal distribution approximated by the combination of two Gaussians.

examples of likely candidates for phonetic clustering because they are CD variants of the same context-independent phone in identical left phonetic context and nearly similar right phonetic context.

Phones that share some phonetic similarity are clustered into sets and yes/no questions are represented as set membership tests. Some questions are generated automatically, e.g. word-position-dependent phones (Is the left phone in the set of word-initial phones?), and it is also possible to manually define linguistic splitting criteria, e.g. "Is the left phone in the set of stressed phones?" or "Is the right phone in the set that denotes stød-bearing phones?". Restrictions on node splitting and merging can also be specified for e.g. silence.

State-tying and phonetic clustering is dependent on the phonetic alphabet used in the phonetic dictionary. Whether the phonetic alphabet is fine, semi-fine or coarse IPA, the number of diacritic symbols such as stress, stød, syllabification or schwa-assimilation can have a significant impact on recognition accuracy (Kirkedal, 2013) and phone classification as demonstrated in Section 4.3.4.1.

#### 2.5.3.2 Descriptive power

The reason GMMs have been successfully applied to acoustic modelling is that the data in both tied and untied HMM states cannot be assumed to be normally distributed. Parametric models such as support vector machines and L1 and L2 regularisers assume that input data follows a normal distribution and data transformation, scaling and normalisation is applied to make the input data more normally distributed. GMMs can model non-normally distributed data by approximating a distribution with a mixture of Gaussians.

The ability of the AM to describe acoustic data is correlated with the number of Gaussians and states it can use to model the data. A high number of Gaussians in each state means the AM can accurately model the training data. A parallel can be drawn to using higher order polynomial features to estimate a more complex decision surface in linear classifiers. The modeling of a non-normal distribution using a mixture of Gaussian components is illustrated in Figure 2.11.

Selecting the right number of Gaussians and states has an impact on performance. The descriptive power of the AM is insufficient to model the data if too few states or Gaussians are estimated. Too many Gaussians will result in overfitting, where Gaussians with nearly identical mean and variance model the same data. Decoding will also slow down if the contribution of a large number of Gaussians has to be estimated and combined by the AM. An excess amount of states can lead to data sparsity, but state-tying can alleviate the problem.

#### 2.5.3.3 Variation and confusability

If several pronunciations of the same words exist in a language due to e.g. dialectal variation, several phonetic representations can be associated to a single word. Figure 2.4 shows an excerpt from a phonetic dictionary. There are two *pronunciation variants* in the dictionary for the name *Svend* highlighted in blue.

| | |
|---|---|
| Svend | s v e n |
| Svend | s w e n |
| ligger | l ɛ g ə |
| lægger | l ɛ g ə |
| på | p ɔ |
| stranden | s d ɾ a n ə n |

Table 2.4: A phonetic dictionary with pronunciation variants and homophones. For illustrative purposes, no diacritics have been assigned to the transcriptions.

Adding pronunciation variants for all words in a dictionary can also decrease recognition accuracy if the pronunciation variants increase the *confusability* of the model. The phonetic representations highlighted in red are identical yet represent two different words.[15] This adds confusability because it is a 1-to-many mapping from phonetic symbol sequence to words rather than 1-to-1 and the number of word sequences the LM has to evaluate at runtime increases. Reducing confusability in the phonetic dictionary to a minimum is a way to reduce the set of word sequences the LM must evaluate.

Another challenge in the mapping from phone sequence to word sequence is the absence of word boundaries in spoken language input. The consequence of missing word boundaries is that all possible word sequences are constructed from the recognised phone sequence. This is illustrated in Figure 2.12, where one phonetic sequence can be translated into a least three different word sequences. This generalises to compound words and sentence boundaries.

[15] Otherwise known as homophony.

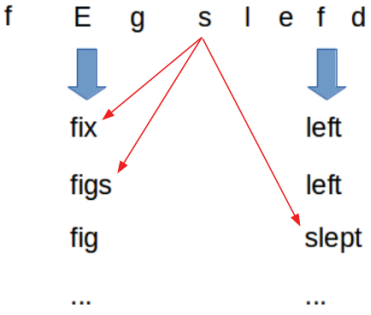



Figure 2.12: Many word sequences can be produced if [s] can be affixed to words both left and right.

If an utterance contains a word that is not in the pronunciation model, it is essentially impossible to recognise. Out-of-vocabulary (OOV) words are frequent in languages with complex morphology such as Latvian (Salimbajevs & Strigins, 2015) or languages with productive compounding like German and Danish.

### 2.5.4 Language Models

In ASR, a standard LM is a Markov model that associates a probability to a word sequence. An illustrative example is to assign probabilities to a sentence such as *Boil water for X* and replace *X* with a word. If *coffee* is inserted for *X*, the sentence makes semantic sense and should be assigned a high probability by a LM. If *guitar* is inserted as *X*, the sentence is nonsensical. In a well-trained English LM, the inequality $P(\textit{Boil water for coffee}) > P(\textit{Boil water for guitar})$ should hold and for this reason, statistical LMs are an effective tool to model sequences in natural language.

The intuition behind statistical language modelling is that a word at position $w_i$ can be predicted from the words at positions $w_{i-1}, w_{i-2} \dots w_1$. For long sentences, this is intractable to compute and in practice only sequences of $w_{i-1} - w_{i-N}$ (known as n-grams) are used.[16] Because the number of n-grams increase exponentially with the size of $N$, $N \leq 4$ is most common. The chain rule of probability is used to assign a probability based on the probability of the words in the phrase.

N-gram LMs can be compactly represented as weighted finite state automata (Mohri et al., 2008). Given a word sequence in the input tape, the weighted automata will return a probability of the sequence of the words (if the sequence and the words are in the language the finite state automata accepts). A weighted

[16]The Markov assumption: The next word depends only on the current word.

finite state automata can be converted to a weighted finite state *transducer* where the same word is on the output tape as the input tape. Each transduction is weighted by the word sequence probability.

This trick makes it possible to perform speech decoding in a finite state transducer framework because we can add a language model to a decoding graph or lattice using finite state composition.

### 2.5.5 Decoding graph construction

The models described so far has been set into a training context by Figure 2.6, but can also be understood in a generative manner where the AM generates feature vectors, the phonetic dictionary generates phones and the LM generates words. The generative view is useful because we can use it to decompose the search for the most likely utterance $\hat{W}$ into two probability models, i.e. the acoustic model and the language model, using Bayes theorem:

$$P(W|O) = \frac{P(O|W)P(W)}{P(O)} \tag{2.3}$$

where the denominator $P(O)$ is usually left out of the equation because it is a constant. The decomposition is necessary because it is not possible to model $P(W|O)$ directly because that entails assigning a probability every possible $W$ which is intractable. However, we now model the speech production mechanism and how 0 is generated. When recognise unseen speech, the decoding process can be understood as a series of translation tasks where each blue arrow in Figure 2.13 is a translation task. The last step may be more adequately described as a reranking task than a translation task.

In practice, and because monotonic alignment can be assumed in speech, the LM, phonetic dictionary, phonetic context-dependency and HMMs can be represented as weighted finite state transducers (WFST) (Mohri et al., 2008) as shown in Figure 2.6. Each transducer can be combined to create a decoding graph and the composition starts from the text side which makes it is possible to determinise and minimise each transducer before computing the next composition.

A LM in Arpa format is converted to a weighted acceptor, which is a type of WFST where the same symbols are on the input tape and the output tape. This WFST is denoted $G$[17]. $G$ is composed with the phonetic dictionary or *lexicon* WFST $L$ using finite state composition. Before the composition operation, determinisation and minimisation are applied to $G$. In $L$, the output symbols are words and the input symbols are phones. Determinisation and minisation remove redundant paths in the WFST graph, which reduces the recognition time, the size of the graph and makes the composition more efficient. The composition $L \circ G$ produces the WFST $LG$.

[17]Stands for *grammar* due to historical reasons, but refers to a statistical LM.

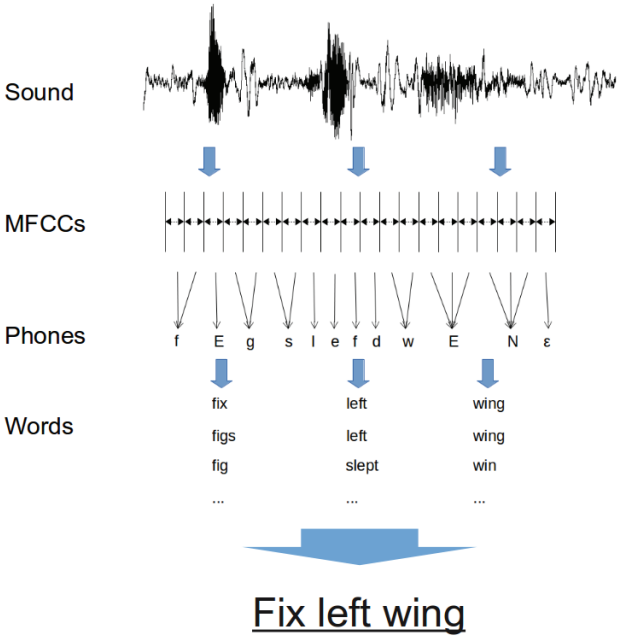



Figure 2.13: A simplified overview of the decoding process. Only a single sequence of context-independent phones is shown to simplify the presentation, though it should have been a lattice with many phone sequences. In the generative view, the arrows are reversed.

Another WFST denoted $C$ represents phonetic context-dependency and is constructed based on the PDT. The output symbols are context-independent phones and the input tape contains CD phones. The same method of composition, determinisation and minimisation is applied to $C \circ LG$. This composition prunes away impossible states created in the PDT stage (Allauzen et al., 2004).

The Kaldi decoding graph recipe also composes the $H$ transducer. As in the conventional recipe, the $H$ output tape is CD phones and the input tape symbols are HMM states or tied states.

The decoding graph creation can be summed up as:

$$HCLG = asl(\min(rds(\det(H' \circ \min(\det(C \circ \min(\det(L \circ G))))))) \tag{2.4}$$

where the $rds()$ operation stands for remove-disambiguation-symbols and $asl()$ for add-self-loops. Dis-

ambiguation symbols are added to the end of phone sequences in $L$ if the sequence is also the prefix of another phone sequence or if two words share the same phonetic representation. The disambiguation symbols ensure that $L \circ G$ can be determinised. Self-loops refer to the arcs in Figure 2.10 on page 29, which transitions from e.g. $B$ to itself which are added in the final outer step of Eq. 2.4.

### 2.5.6 Decoding

Decoding is the process of finding the most likely word sequence that was spoken in an utterance. This is modelled in the fundamental ASR equation which combines the probabilistic information from the LM and AM to find $\hat{W}$:

$$\hat{W} = \underset{W \in LANG}{\arg\max}\ P(O|W)P(W) \tag{2.5}$$

where $LANG$ are the utterances in a language, $W$ is a single word sequence (or utterance), $P(O|W)$ is the probability assigned by the AM and $P(W)$ is the probability assigned to the word sequence by the LM. The *decoder* that finds the most likely sequence is represented by the arg max operation.

Equation 2.5 can be extended with pronunciation model probabilities, term weights and log-likelihoods to get the extended ASR equation:

$$\hat{W} = \underset{W \in LANG}{\arg\max} \Big[\lambda_1 \log P(O|W,P) + \lambda_2 \log P(P|W) + \lambda_3 \log P(W)\Big] \tag{2.6}$$

$P(O|W,P)$ can be reduced to $P(O|P)$ because phones are not dependent on words and E. 2.6 becomes:

$$\hat{W} = \underset{W \in LANG}{\arg\max} \Big[\lambda_1 \log P(O|P) + \lambda_2 \log P(P|W) + \lambda_3 \log P(W)\Big] \tag{2.7}$$

where $\lambda_1$ is known as the *acoustic scale weight* and $\lambda_2$ as the LM weight. $\hat{W}$ will depend on the relative contributions of the terms in Eq. 2.7. Because the phonetic representation of short function words are sub-sequences of longer words, it is necessary to constrain the influence of the AM on $\hat{W}$ relative to the LM. Otherwise, $\hat{W}$ will mainly consist of short words. This constraint can be implemented by adding a Word Insertion Penalty to Eq. 2.7, but the hypothesis generation can also be constrained by ensuring that $\lambda_2 >> \lambda_1$.

The decoder chooses the most likely word sequence $\hat{W}$ from a set of word sequences. In practice, the decoder traverses the decoding graph to find the most likely path. Exhaustive search of the decoding graph is intractable and three parameters constrain the possible traversal of the decoding graph:

1. Decoding beam

2. Lattice-beam

3. Max active states

To understand the point of these parameters and their interaction, imagine first that we traverse the decoding graph with beam search to the end of an utterance. The *beam* parameter determines how we traverse the graph to produce a *lattice* which is a compact representation of the n-best ASR hypotheses. Subsequently, paths in the lattice are discarded if the cost of the paths are not within the *lattice-beam* of the cost of the best path. Rather than decoding an utterance until the end and removing unlikely paths, the lattice-beam is applied in periodical backward sweeps during decoding. The *max-active-states* threshold limits the number of active states at each time frame such that only the most likely paths are explored. The most likely paths or active states can be found using a histogram of likelihood scores and is also known as *histogram pruning*. The motivation of the parameter is that only the most likely states at each step in the decoding are likely to survive beam pruning in subsequent steps in the lattice. This type of pruning has the benefit of defining a worst case processing time for decoding.

### 2.5.7 Medical dictation scenarios

The two medical dictation scenarios – real-time ASR, and ASR and post-editing – require two different systems. Real-time ASR requires an online ASR system that decodes in real-time and output results incrementally. *Online* ASR systems receive speech input from a microphone where *offline* ASR systems take recordings as input. *Incremental* ASR systems output recognised words or phrases as an utterance unfolds in time, whereas *End pointing* is when ASR systems output recognition results after the utterance is finished. The ASR system in the ASR and post-editing scenario can therefore be a slower-than-real time, end-pointing and offline ASR system.

The system type has an impact on the recognition rate. Online and offline systems work identically, both can do end-pointing or incremental decoding and both can decode in real-time. If decoding is under a real-time constraint, then the real-time factor (RTF) needs to be below 1. To obey $RTF < 1$ usually means restricting the size of the lattice and narrowing search beams via the three decoder parameters from Section 2.5.6. If the size of the beam and lattice is smaller, decoding speed increases but the likelihood that a correct ASR hypothesis is found during decoding becomes smaller.

If recognition results are incrementally output, less lexical context in the LM can be used to traverse the graph and it is less likely that the best path in the decoding graph is found. It is possible to use a larger context if an ASR system uses end-pointing, but waiting until the end of a sentence to see the recognition results is usually undesirable in interactive systems and transcription because of the delayed response. If

a word is falsely recognised at the beginning of a sentence, a user would have to wait until the end of the sentence before noticing the error and have to repeat the entire sentence, but with incremental output, the user can stop after a few words.

## 2.6 Discussion

We do not know exactly what stød is in acoustic terms, i.e. there is no single acoustic measure that we can extract as a good predictor of stød. There are a number of acoustic measures may that may correlate with stød but the literature agrees on a few indicators such as

1. Disharmony (aperiodicity)
2. Decrease in intensity
3. Decrease in pitch

If stød can only occur where there is stødbasis, the same acoustic events that would signal stød on one phone are meaningless when occurring on another phone. Based on these observations, it should be questioned whether stød is an acoustic event that should be considered separately from segments or whether e.g. [ɒ] should be treated as different from [ɒˀ]. If treated separately, late integration of phonetics and prosody would be required to take advantage of the lexical function of stød.

Detecting stød by itself is interesting from an academic point of view. The question whether stød is in fact one or more different phenomena, as suggested in Hansen (2015), that signals the same lexical distinction could be investigated and separating features which signal stød from the stød-bearing phone will give insight into the nature of stød. Feature interactions could also provide new information that can characterise stød phonetically.

If the task is to characterise stød, it is necessary to treat stød as separate from the stød-bearing segment. Stød can cross segment boundaries and the acoustic events that signal the manifestation of stød can be realised or not irrespective of a segment and realised differently if stød can be manifested in different ways as Hansen suggests. If the task is to be able to discriminate between stød-bearing and stød-less segments, stød detection using a classifier as an indicator function or to output a probability estimate as input feature to a classifier would be one approach to handle this task. Such a feature could aid a phone classifier to discriminate stød-bearing and stød-less phones.

Another approach is to jointly classify segment and stød. In place of a single binary or logprob feature output by a stød classifier, the input features to the stød classifiers can be directly used to extend the feature

vector used as input to the phone classifier. From an application point of view, the two approaches achieve the same goal and either could be used as a step on the path to phonetic characterisation. Appropriate interrogation of the classifier could give insights for further study or, if the phone classification error can be minimised, the classifier can annotate data for phonetic investigation.

The application scenario is important for the choice of ASR system. If decoding speed is important for the application, restrictions on the ASR system can potentially reduce ASR accuracy, but if the ASR system can use end-pointing, large lexical context and traverse a larger part of the decoding graph, it is more likely that a correct ASR hypothesis can be found. If we cannot model large lexical context because we do not have access to data, the best way to improve ASR is to improve acoustic modelling, which makes stød modelling in ASR a relevant and interesting research area.

# Chapter 3

# Annotation study

This chapter is devoted to an investigation into the reliability of phonetic annotation by experts and we will use the terminology introduced in Chapter 2 extensively. Annotated data is a prerequisite to conducting quantitative investigations of stød. To be able to train statistical models to detect stød, the reliability of the annotation of stød is vital. It is, however, not an easy task to annotate audio, and especially not stød – especially because stød is not used by some speakers and it is perceptual in nature. While a layperson can hear the difference between *viser*/*viser* (verb/noun) or *bønner*/*bønder* (EN: beans/farmers), contributing the perceived difference to the realisation of stød requires training in phonetics. Annotators must be trained to be able to create phonetic annotations and expertise is necessary to annotate prosodic features such as stød, stress, etc.

## 3.1 Annotation reliability

As described above, stød is not a well defined phonetic unit from a computational linguistics point of view. Corpora with stød annotation are primarily designed for human consumption and manual analysis. This presents problems if the resources are to be used for computational experimentation. Because stød and other suprasegmental features affect more than the phone they are annotated on, the phonetic annotation requires additional interpretation by a linguist to estimate the time domain in which a suprasegmental feature is realised. The interpretation varies highly with the perceptual capabilities of the annotator, the interpretation of the labelling schema and other annotator characteristics and finally the difficulty of the annotation task (Gut & Bayerl, 2004).

Therefore, if sufficient data for computational experimentation can be found, the question remains whether this annotation is reliable. Because there are many nuances to consider (acoustic realisation, start

or end of a sound or prosodic feature to name a few), the annotation of stød can be considered a difficult task that can only be handled by trained phoneticians. Crowd sourcing e.g. via Amazon Mechanical Turk is not appropriate for this task (Novotney & Callison-Burch, 2010).

### 3.1.1 Annotators

Even if stød is annotated by experts, can the annotation be relied upon? Phonetic and prosodic annotation relies to a great extent on annotator experience, hearing and subjective evaluation. If expert annotation can be found, whether an annotation guideline existed and whether it was followed can be difficult to ascertain.

### 3.1.2 Ground truth

Phonetic annotation lacks a gold standard or *ground truth* for comparison and evaluation. While canonical phonetic or phonological transcriptions do exist in e.g. dictionaries, comparison to colloqial spoken language is an inadequate solution. Participants can be asked to pronounce the same utterances, but inter-speaker variability will make each pronunciation of the same utterance different. The particular pronunciation of an informant will be influenced by dialect, sociolect, age, gender, time of recording, etc., and none of these influences are taken into account in canonical transcription such as those in large general-purpose dictionaries.

## 3.2 Experimental setup

In corpus linguistics, it is common to compare annotation by several annotators when evaluation against a gold standard is not possible. *Inter-annotator agreement* can be used to give an indication of whether a transcription or annotator is reliable.

Inter-annotator agreement compares label sequences between annotators and the intuition is that labels which show a high degree of similarity are more likely to be correct. The label sequences need to use the same label set[1] e.g. word classes or phones to be comparable.

In the description of this experiment, key terms are *label* and *item*. Depending on the purpose of a phonetic annotation, several linguistic levels can be mixed together in the same annotation. Thus an annotation can contain segments, suprasegments, phones, tone or diacritic symbols in the same annotation string. To collectively refer to all the symbols in an annotation, a transcription is referred to as a label sequence and the term *label* is used to refer the symbols in that label sequence whether they are segments, phones, diacritic symbols etc.

[1] At least, different label sets must be deterministically mapped to each other.

The object denoted by a label is referred to as the item, so a phonetic symbol (the label) labels a speech sound (the item).

### 3.2.1 Data

The data used for the reliability study is an interview with a high school student. There are two participants in the recording (the interviewer and the interviewee), and their speech has been phonetically transcribed by four expert phoneticians from University of Copenhagen. The phonetic transcriptions were manually aligned with orthography and the four transcriptions were manually segmented to phone level. The recording is 98 seconds long with one minute of annotation starting 16.6 seconds into the recording. The corpus is designed and created by Jan Heegård Petersen and will be referred to as the JHP sample.

One minute is a small sample and we use it because it is the only data we were able acquire with four expert annotations. Phonetic annotation of spontaneous speech takes approximately a factor of 38 (Li et al., 2000) times four (1 minute audio = 2.5 hours transcription) and subsequently the annotations need to be aligned by another expert, i.e. a time-consuming, labour-intensive and costly process. To maintain the high quality of the data, we decided to use the sample as is rather than acquiring more data from e.g. student.

The corpus was annotated using Praat (Boersma, 2002) and contains only annotation for orthography for both participants when presented to annotators. The annotators were asked to transcribe the sample using *semi-fine* IPA (Grønnum, 2005). Praat uses `Tier` objects to manage label sequences. A tier contains a sequence of `Interval` objects which contain a label such as phone. `Interval` objects are time-coded with a start time and an end time that cannot overlap with other intervals in the same tier, so we know the duration of a speech sound. All tiers are contained in a `TextGrid` which is illustrated in Figure 3.1.

The label sequences that we compare are denoted IPA1-segment, IPA2-segment, IPA3-segment and IPA4-segment and IPA1, IPA2, IPA3 and IPA4 identify the four annotators. The experts disagree on stød labelling at the end of the annotation sequences in Figure 3.1. IPA1 and IPA4 label stød on the same item, while IPA2 assigns a stød label to the previous item. The disagreement could be a caused by stødbasis because IPA2 uses a label with a long vowel ([æːˀ]) unlike IPA1 and IPA4. Also, stress annotation ([ˈ] and [ˌ]) is treated as a label rather than a suprasegmental feature and the use of diacritics produce labels in one label sequence that may not appear in another sequence in similar fashion to [æˠ] only occurring in one transcription of *fremtrædent* (IPA1-segment).

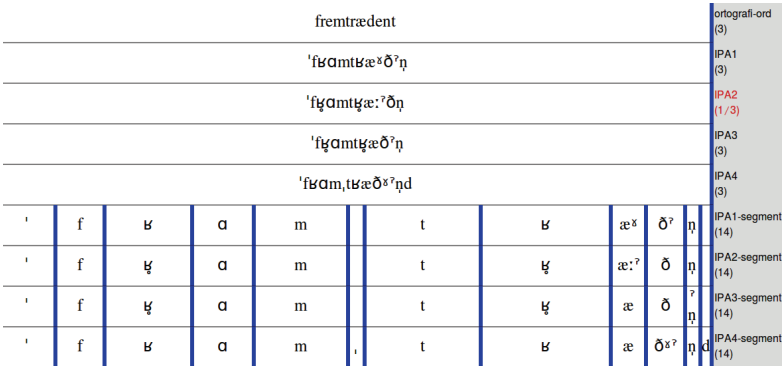



Figure 3.1: Danish word *fremtrædent* (EN: prominent) in the JHP sample. Only tiers relevant to the experiment are shown and e.g. phonological and parts-of-speech annotation have been removed from this image.

### 3.2.2 Method

The simplest way to compare label sequences that are monotonically aligned is *majority voting*. Majority is defined in terms of label agreement, i.e. how many annotators agree on a label and we sum the label agreement for all labels.

In this study, different majority thresholds are investigated. The threshold might be decisive in the inter-annotator comparison. We define three definitions of majority ($N_{agree}$) with 4 annotators:

**All** $N_{agree} >= 4$

**Major** $N_{agree} >= 3$

**Tie** $N_{agree} >= 2$

**Any** $N_{agree} >= 1$

Using these definitions, we compare the label sequences and their agreement. To give an indication of the number of items that could potentially carry stød, **Any** is used as a baseline for comparison.

#### 3.2.2.1 Label sets

The interpretation of *semi-fine* IPA is different from annotator to annotator. The difference is often demonstrated in varying use of labels. The consequence is a substantial difference in the label sets used because

there are many possible combinations of diacritics and phones. Each annotator will use a subset of the possible combinations and while each label set will overlap with other annotators, the size of each set will differ. In this corpus, a total of 178 labels are observed, but any single annotator uses at most 107 labels. All annotators have only 55 labels in common.

For statistical comparison, a set of equivalence classes or a mapping to a more coarse and commonly used label set can alleviate the problem. Because the study is focused on the reliability of stød annotation, a binary label set consisting of +/- stød is also compared.

Because reduction to a binary task can be too harsh an application of Occam's razor, we compute more advanced comparisons of the inter-annotator agreement. In addition to raw counts and frequency analysis, Cohen's $\kappa$ (Cohen, 1960) is a widely used inter-annotator agreement statistic that is corrected against chance agreement. The formula is:

$$\kappa = \frac{Pr(a) - Pr(e)}{1 - Pr(e)} \tag{3.1}$$

where $Pr(a)$ is the relative observed agreement and $Pr(e)$ is the probability of *agreement by chance*. $Pr(e)$ is calculated as $\sum_c^C P(c)$ where $P(c)$ is the expected random agreement for label $c$ and $P(c) = \prod_a^A P_a(c)$ where $P_a(c)$ is the probability of annotator $a$ assigning label $c$ to an item.

The final item in Figure 3.1 is an example of agreement by chance. The item is labelled [d] by IPA4, but not labelled by any other annotator. In this case, 3 annotators agree, but only as a side-effect of the alignment or by chance.

$\kappa$-scores above 0.6 represent adequate agreement, while 0.8 and above indicate almost perfect agreement. Agreement worse than chance is indicated by 0 and 1 is perfect agreement. $\kappa$ can become negative for cases where there are systematic disagreement (Nowak & Rüger, 2010).

Cohen's $\kappa$ measures pairwise inter-annotator agreement between two annotators but can be averaged over pairs of annotators. The result is a metric that can be interpreted similarly to standard $\kappa$, but for multiple annotators.

For tasks with only few labels, e.g. binary classification, Cohen's $\kappa$ can be low though the raw agreement is high, because $\kappa$ overestimates $P(e)$. This problem is referred to as the paradox of $\kappa$ (Feinstein & Cicchetti, 1990). In agreement based statistics, two wrong, but identical label assignments result in agreement. This is especially problematic for skewed data sets where prevalence is high for one class (Passonneau & Carpenter, 2014), which is the case for the JHP sample where 94% annotations are without stød.

To control for these effects, Multi-Annotator Competence Estimation (MACE) (Hovy et al., 2013) is used to calculate annotator competence. MACE is designed to identify reliable annotators in adversarial conditions. Adversarial conditions refer to cases where an annotation task is crowd-sourced e.g. via Amazon

Mechanical Turk and a subset of annotators are not trying to create the correct label sequence. A small payment is usually given for completed annotation and this has given rise to *spammers* that will annotate at random or use the same label for all annotations. Reliable annotators can be identified by comparing label sequences to a gold standard, but in the absence of a gold standard, different measures of inter-annotator agreement are used. The intuition is that reliable annotators will correlate with each other, while spammers will stand out because of their randomness.

MACE uses an item-response model to calculate annotator competence rather than an annotation metric. The underlying model assumption is that the annotator will produce the correct label sequence if he tries to. In this case with only expert annotators, the assumption should be valid. The key difference between $\kappa$ and item-response models is that the former has a focus on annotators while the latter has a focus on the *item*. Using Pearson's $\rho$, the authors show that the annotator competence metric has a higher correlation with annotator proficiency than $\kappa$.

Annotator competence on both full and binary label set is computed. The competence estimate is interesting in both tasks, but the statistic will be especially informative in the binary task together with majority voting and $\kappa$ (Hovy et al., 2013).

### 3.2.3 Analysis

**Majority voting** The definition of majority in the majority voting scheme has a large impact on the agreement measure as can be seen in Table 3.1. Agreement on the amount of stød annotations fluctuate between 78% ($N_{agree} >= 2$) and 55% ($N_{agree} >= 4$) out of 78 possible stød assignments (according to *Any*-majority). More informative statistics are necessary to evaluate the quality of the transcriptions.

| Majority | Any | Tie | Majority | All | Total #labels |
|---|---|---|---|---|---|
| #stød | 78 | 61 | 50 | 43 | 995 |

Table 3.1: Number of stød annotations using different majority definitions: *Any*=1 annotator annotates for stød, *Tie*=2 annotators agree on stød, *Majority*=3 annotators agree, *All*=4 annotators agree.

**Inter-annotator agreement** Tables 3.2 and 3.3 are pairwise Cohen's $\kappa$ confusion matrices.

When evaluating on the full label set, there are 178 observed labels and 995 items. All pairwise comparisons as well as the per-annotator average $\kappa$ are above the lower bound of 0.6 for adequate annotation. The mean agreement[2], 0.74, is also significantly above the lower bound. With a standard deviation of 0.02,

[2]Average of Avg. $\kappa$.

| | IPA1 | IPA2 | IPA3 | IPA4 |
|---|---|---|---|---|
| labels | 107 | 94 | 99 | 107 |
| IPA1 | 1.00 | 0.69 | 0.74 | 0.74 |
| IPA2 | 0.69 | 1.00 | 0.75 | 0.76 |
| IPA3 | 0.74 | 0.75 | 1.00 | 0.78 |
| IPA4 | 0.74 | 0.76 | 0.78 | 1.00 |
| Avg. | 0.72 | 0.74 | 0.76 | 0.76 |

Table 3.2: Inter-annotator agreement confusion matrix calculated with Cohen's $\kappa$ on the JHP sample. The basis for this matrix is all observed labels. *#labels*=the number of labels used by that annotator.

only IPA1 is more than one standard deviation below the mean. This is high agreement considering that any single annotator only uses at most 60% ($\frac{107}{178} = 0.601$) of the full label set and that annotators only have 55 labels in common.

The reason can be seen in Figure 3.2. The values on the x-axis are raw agreement counts per item. The raw count is 0 if the annotator assigned a label none of the other annotators used for a given item and 3 if all annotators assigned the same label. The bar plots are different from the confusion matrix in the respect that they compare one annotator to all other annotators per item instead of making a global and a pairwise agreement comparison.

The graphs suggest that there is a prevalence of a subset of labels and that the annotators agree on these labels. In such a scenario, it is likely that the disagreements are few and not systemic. The plots for IPA2, IPA3 and IPA4 all show a Zipfian tendency. We assume that the label distribution depends on word distribution, i.e. the distribution of labels will change if word distribution changes, but it does not follow that the label distribution should be Zipfian. It seems reasonable to attribute the difference to pronunciation variation and inter-annotator disagreement.

The distribution for IPA1 is not Zipfian. IPA1 assign different labels than other annotators for approximately 50 items. This is reflected in the $\kappa$ statistics in Table 3.2 where IPA1 receives the lowest pairwise and average agreement scores.

To investigate the assumption that annotators agree on a small subset of highly frequent labels, the label frequency histogram is plotted in Figure 3.3a. Indeed, there is a small number of prevalent labels and as can be seen from Figure 3.3b, the annotators agree to a high extent on this small subset. The only differences

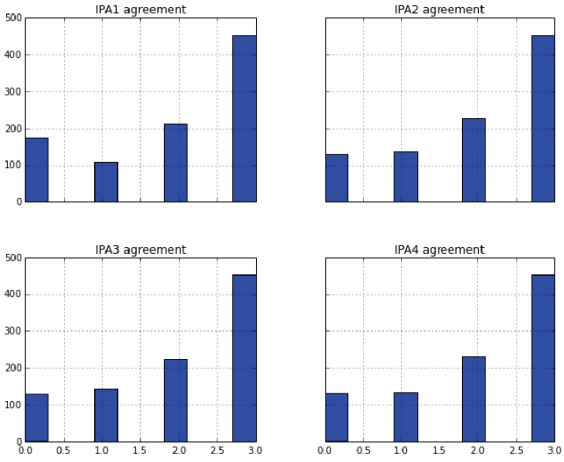
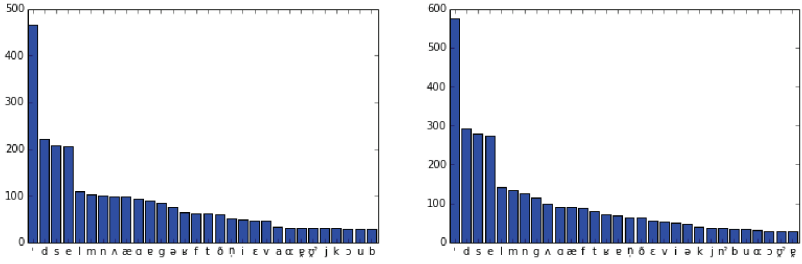



Figure 3.2: Pairwise label agreement by annotator.

between the plots in Figure 3.3 are the missing [a] from 3.3b and [n$^?$] from 3.3a and the sequence of the labels and label pairs.

Another result that can be extracted from the plots in Figure 3.3 is that the annotators only agree on one or two labels containing stød. The raw agreement counts for stød in 3.3b is 36 and 29. Note that the method of counting used here and in Table 3.1 is not directly comparable. Majority is defined using >=. As a result, an item with 3 identical labels would count as 1 in majority voting using $N_{agree} >= 2$ but as two agreement pairs because annotator $A_1$ used that same label as $A_2$ and the same label as $A_3$. This difference is important to not be deceived into believing that the two highly frequent pairs make up 90% of the stød agreement in Table 3.1. Applying majority counting to the stød labels in Figure 3.3b gives an agreement of 6 and 11.

**Binary labelling** While it is a positive indication that two labels containing stød are among the 30 most frequently used, it is not a strong enough indicator that stød annotation is reliable. To further investigate the reliability of stød annotation, the agreement of the binary label set, which only considers stød, is studied next. This filtering is motivated by annotators disagreeing only on a segment, but not on stød annotation, which is also the case in Figure 3.1.




(a) Label frequency

(b) Label pair agreement frequency

Figure 3.3: Histograms of raw label frequency (a) and raw label pair agreement frequency (b) across all annotators. Two assignments of the same label to an item is one agreement pair. Histograms only display the 30 most frequent labels.

The label sequences IPA1-segment, IPA2-segment, IPA3-segment and IPA4-segment are binarised, e.g. [æːˀ], [ðˀ] [ðːʏˀ] and [ˀn] in Figure 3.1 become 1 and the remaining labels become 0.

A side-effect of this filtering is a skewed data set. The non-stød class represents 92%-96% of labels assigned[3] and would not produce meaningful statistics for the interpretation of reliability of stød annotation. Again, the chance agreement correction of $\kappa$ becomes important for the trustworthiness of the statistical analysis.

The agreement scores for the binary label set in Table 3.3 are even higher than the $\kappa$-scores in Table 3.2. This is expected as a labelling task with two labels is easier than a task with 178 (observed) labels, even though prosodic annotation is a very difficult task. The average agreement is above 0.8 for all annotators and the mean agreement is above 0.8 at 0.82. The standard deviation is 0.02 and again IPA1 is more than one standard deviation below the mean.

Figure 3.4 illustrates the background for the $\kappa$-scores. To reduce skewness, the analysis ignores items which have not been labelled with stød by any annotator. Agreement statistics for the remaining items will only focus on the agreement of stød annotation. As expected from Table 3.1, the annotators agree completely on 43 label assignments. There is a low number of midrange disagreements and 20-27 label assignments per annotator where they do not agree with any other annotator.

**Error analysis** The high number of disagreements contradict the hypothesis that stød annotation as reliable and warrants manual investigation. I discovered that off-by-one errors in the alignment are frequent. In 10 cases, the assignment of stød labels are off-by-one. In Figure 3.1, an alignment error is visible in the

[3]By majority counting.

| | IPA1 | IPA2 | IPA3 | IPA4 |
|---|---|---|---|---|
| #stød | 53 | 58 | 62 | 59 |
| IPA1 | 1.00 | 0.78 | 0.84 | 0.85 |
| IPA2 | 0.78 | 1.00 | 0.75 | 0.88 |
| IPA3 | 0.84 | 0.75 | 1.00 | 0.83 |
| IPA4 | 0.85 | 0.88 | 0.83 | 1.00 |
| Avg | 0.82 | 0.80 | 0.81 | 0.85 |

Table 3.3: Inter-annotator agreement confusion matrix calculated with Cohen's $\kappa$ on the JHP sample. The basis for this matrix is binary +/- stød labels. *#stød*=the number of stød annotations made by that annotator.

labelling of the second to last interval in tier IPA3-segment ([ˀn̥]). It is an error because stød is prefixed to the phone [n̥] rather than a suffix. This is not a phone or segment according to any definition of IPA known to the author and not a well-formed label. As is reflected in the transcription of the entire word in tier IPA3, stød should have been affixed to the previous phone.

Additional examples where stød labels are off-by-one can be seen in Figure 3.5. The discrepancy can be caused by genuine disagreements or be due to the different interpretations of semi-fine IPA annotation. In 3.5a, the annotators disagree on whether to label the sound they heard as [æːˀ] or [æɪ̯ˀ]. Similarly in 3.5b, the annotators disagree on [oːˀ] vs. [oɐ̯ˀ]. In 3.5c, the disagreement also stems from whether to use [yː], the long version of the vowel, vs. a combination of [y] and [ɐ̯]. While IPA1 and IPA3 most often uses two labels

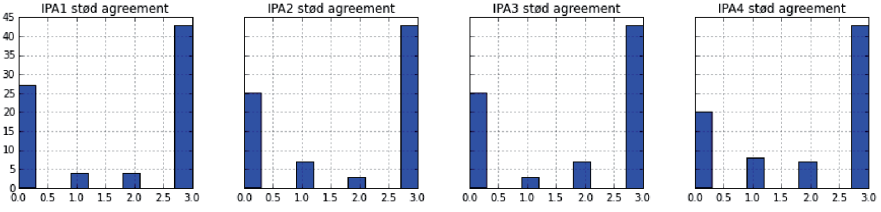



Figure 3.4: Pairwise stød label agreement per item by annotator.

with stød on the last sonorant instead of a long vowel, IPA2 and IPA4 prefers to assign a long vowel with stød, though on one occasion, IPA2 opts for two labels.

| gymnasier | tror at | bestyrelsen |
|---|---|---|
| gymˈnæɪ̯ˀɕə | ˈtʁoɒ̯ˀad | beˈsdyɒ̯ˀˌlsn̩ |
| gyˈnæːˀsɐ | ˈtɤ̯oːˀæd | beˈsdyːˀɐlsn̩ |
| gymˈnæːˀɕə | ˈsʁoɒ̯ˀˈad | beˈsdyɒ̯ˀˌlsn̩ |
| gynˈnæːˀɕə | ˈtʁoːˀæd | beˈsdyˀɒ̯lsn̩ |

(a) EN: High school (b) EN: Think that (c) EN: The board

Figure 3.5: Examples of off-by-one alignment errors.

As mentioned in Section 2.1.1, prosodic features do not only affect the segment they are affixed to, but also the phonetic context. The scope of prosodic features are in fact syllables rather than segments and it is possible that stød influences both segments. However for statistical purposes, a discrepancy such as this will give misleading results and the notational variation must be addressed.

In all off-by-one examples found by the author, the reason for the discrepancy was either an alignment error as in Figure 3.1 or due to the annotator label choice for the vowel in a syllable. Irrespective of the label choice of the annotators, stød is annotated on the nucleus of the same syllable.

**Correction** We create a modified version of the stød annotation where alignment errors were corrected and cases where stød was annotated on the same syllable, as in Figure 3.5, were aligned to each other. These modifications entailed all 10 cases found during manual inspection. The corrected pairwise stød label agreement is shown in Figure 3.6.

Correction produced a different picture than was painted by Figure 3.4. A significant reduction in disagreements, i.e. where an annotator labelled an item with stød and no other annotators did, and an expected increase in total agreement pairs is observed. There is a clear indication of reliability in stød annotation.

**Competence** This is furthermore indicated by the annotator competence statistics in Table 3.4. It is also clear that competence statistics for the original label set and the corrected stød labelling are correlated.

*Competence phone* uses the observed phonetic annotation as labels and *Competence stød* refers to the binary labelling corrected for alignment errors.

| Annotator | #labels | Competence phone | #stød | Competence stød |
|---|---|---|---|---|
| IPA1 | 107 | 0.760 | 53 | 0.770 |
| IPA2 | 99 | 0.813 | 58 | 0.840 |
| IPA3 | 94 | 0.823 | 62 | 0.894 |
| IPA4 | 107 | 0.833 | 59 | 0.856 |

Table 3.4: Annotator statistics on the JHP sample computed with item-response models trained with MACE.

### 3.2.4 Chapter conclusions

Based on the analysis in Section 3.2.3, several conclusions on phonetic annotation and prosodic stød labelling can be drawn.

Based on $\kappa$-scores and annotator competence statistics, we conclude that the phonetic annotation is reliable. The reliability is challenged by the subjective nature of phonetic annotation tasks, different label sets as a consequence of different interpretations of loosely defined annotation guidelines and a restricted common label set. However, because annotators agree on highly frequent labels, they achieve high $\kappa$ scores.

Based on manual error analysis of stød annotation, errors in the phonetic alignment become apparent and the problem is caused by the different label sets used by the annotators, subjective interpretation of acoustic events and the lack of a definition of syllables in segmental phonetics that can be interpreted computationally.

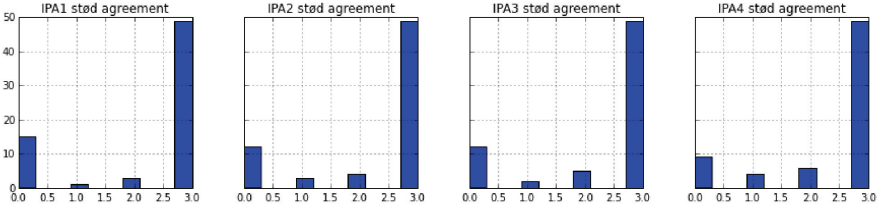



Figure 3.6: Corrected pairwise stød label agreement per item by annotator.

Correcting for alignment errors, $\kappa$-scores and annotator competence statistics indicate a high degree of agreement and in turn points to stød annotation as a reliable source of information if annotated by experts. Because at most 7.8% items are stød-bearing, the agreement and competence statistics are challenged by the skewness of the data set. The chance agreement correction of $\kappa$ and item-response model should correct for this skewness and is supported by the raw agreement counts that only focus on the stød label.

Stød annotation by experts is reliable according to the findings in this chapter. Reliable annotation is crucial for quantitative analysis of stød in Chapter 4 and because stød annotation is reliable, acoustic features can be extracted from sound samples where stød is realised and used to train statistical models. These statistical models can be used in classifiers to detect stød in unlabelled data if salient acoustic features are extracted.

# Chapter 4

# Stød detection

In Chapter 3, a focused study on the reliability of the prosodic annotation of stød concluded that stød labelling is reliable if the annotation has been carried out by phonetic experts. The present chapter will investigate how this knowledge can be used and audio recordings with aligned stød annotation will form the basis of the study. Because stød annotation can be relied upon to be accurate when annotated by experts, the phonetic transcription will indicate when stød is realised in a recording and make statistical analysis of stød feasible. However, sound is a continuous and complex signal that must be decomposed before acoustic features that describe stød can be extracted. Some acoustic features are known to be predictive of stød such as voice quality features, but a broad investigation of stød-correlation has not been conducted. The study performed in Hansen (2015) is thorough, but the data material is restricted to a single male speaker and read-aloud text in lab conditions. Statistical analysis can estimate how salient individual features are for stød detection and use this knowledge to select features to train classifiers to detect stød.

Stød is difficult to describe because it's properties are not clearly defined. As previously mentioned, stød is a prosodic and perceptual feature. The manifestation of one or more of the acoustic correlates of stød will signal stød only when it occurs on a sonorant. This simple working definition of stød is inadequate to describe the actual distribution of stød, but illustrates well a problem for stød detection: it is unknown whether it is possible to detect *only* stød. The known acoustic events that signal stød can occur where stød is not perceived and same cocktail of acoustic signals may not be realised every time a stød is annotated, especially if more than one strategy to signal stød exists (Hansen, 2015). Much research has gone into characterising stød and the working description in Fischer-Jørgensen (1989); Hansen (2015) and several papers by Grønnum & Basbøll reduces to characterising stød as "creak, but more than creak".[1] An explorative investigation of

[1] The author's own paraphrase.

acoustic feature correlation with stød cannot provide an answer, but can indicate areas of research where an answer to the ephemeral "...more than creak" can potentially be found.

If the statistical models are not able to describe stød, detection of stød might still be possible if stød is detected jointly with the stød-bearing sonorant. Such a classifier does not solve the puzzle of "...more than creak", but can detect different classes of stød (stød-bearing phones) if sufficiently accurate. Such a classifier could be applied to annotate larger corpora of spoken data and create a better basis for statistical analysis in addition to being useful to downstream NLP applications.

The start of this chapter is devoted to a description of the data used to estimate statistical models as well as the preprocessing steps, phonetic alignment and feature extraction process. The evaluation of feature salience in Section 4.2 is formulated as a feature ranking experiment using decision tree classifiers to analyse the features introduced in Section 2.4. Section 4.3 presents the stød detection experiments which are formulated as a binary classification task in Section 4.3.3 and a multi-class classification task in Section 4.3.4.

The experiments use the machine learning library *scikit-learn* (Pedregosa et al., 2011) extensively. Scikit-learn is implemented in Python and managed by researchers at INRIA, France. Machine learning algorithms like Logistic Regression, Support Vector Machines (SVM), Naive Bayes (NB), Gaussian Mixture models (GMM) and perceptron are implemented, as well as feature projections such a Linear Discriminant Analysis and Principal Component Analysis (PCA). Feature ranking using tree classifiers and feature selection and feature-space transformations are also possible as well as normalisation, scaling and preprocessing. Data management is handled using a library named *pandas* (McKinney, 2012).

## 4.1 Data

The JHP sample from Chapter 3 is modified and reused in this chapter. The length of the JHP sample is 1 minute 38.54 seconds. The annotation starts 16.67 seconds into the recording and ends 91.95 seconds after the start of the recording. In the annotation study, initial and final silence are counted as 2 items. When audio is sampled at a 10 ms rate, initial and final silence add to the skewness of the test data. Based on start and end times, unannotated parts of the JHP sample are discarded.

Stød support in the JHP sample, i.e. the number of 10 ms samples that are labelled with stød, ranges from 339-554 (depending on the annotator) out of a total 7535 samples, if a 10 ms sampling frequency is applied. Thus 4.5-7.4% of the data is labelled as stød-bearing. This is a very low number of samples and because statistical analysis relies on the law of large numbers, more data is necessary to apply statistical analyses to the acoustic features. Data from three corpora will be used in the experiments in this chapter.

The JHP sample will serve as a test set, while data from DanPASS and DK-Parole will serve as training data. The training data is chosen because it has been manually annotated by phonetic experts and it is the only one of its kind available.

### 4.1.1 Danish Phonetically Annotated Spontaneous Speech corpus (DanPASS)

DanPASS (Grønnum, 2006, 2009) consists of monologues and dialogues of unscripted speech. Only the monologues are used in this experiment and were collected during three separate tasks: two description tasks and a map task.

The first task is a description task. The speaker is presented with a network of geometric shapes and asked to describe the network. The task was designed to reveal whether the speakers look ahead and signal utterance boundaries using prosodic information prior to the boundary.

The second task is a map task where the speaker guides the experimenter through 4 different routes on a city map.

In the last task, the speaker is given a model of a house and the individual building blocks of the house. The speaker describes how to assemble the blocks to resemble the house.

The monologues were recorded in 1996 using a Sennheiser Microphone ME64 in lab conditions and later digitised with a 48 kHz sampling rate. The recorded speech is one-way communication with the experimenter who offered no feedback once instructions were given.

The group of speakers consisted of 13 men and 5 women aged 20-68. They all originate from the Greater Copenhagen area and had no known language deficiencies. The monologues total 2 hours and 51 minutes of speech, 1075 word forms and 21170 running words.

The DanPASS annotation includes orthography, detailed and simplified parts-of-speech and semi-fine IPA annotation at the word and syllable levels. Phonetic annotation was carried out by two annotators separately using Praat and in all, 3 pairs of annotators have been involved. For each file and speaker, the annotation was compared and in cases where the annotators disagreed, Grønnum served as arbiter.

An overall good agreement between annotators is cited as an indication of the validity of the phonetic annotation. With regards to the reliabilty of stød annotation, there is an overlap between the annotators used in DanPASS and the JHP sample.

Because only the monologues are used, the DanPASS sub-corpus will be referred to as DanPASS-mono in subsequent chapters.

### 4.1.2 DK-Parole

DK-Parole (Henrichsen, 2007) contains text from newspapers and recordings of read-aloud speech from a single male speaker. Like DanPASS and JHP, Praat TextGrids contain all annotations. The annotation uses time-coded X-SAMPA transcription and is not as fine-grained as the DanPASS annotation because the granularity of the phonetic transcription is at the word level. The transcription is manual and there is an overlap with the annotators from the JHP sample.

The audio was recorded in 2006 and 2008 at Copenhagen Business School. The speaker was situated in a lab and the recordings are without noise. DK-Parole is much larger than DanPASS-mono, approximately 17 h. To balance the need for additional data and letting a single male speaker dominate the data, a sub corpus of 48 min. was selected randomly from DK-Parole.

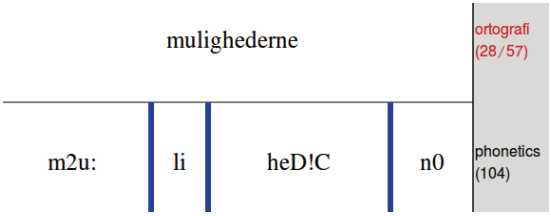
For simplicity, the 48 min DK-Parole sub-corpus will be referred to as Parole48 henceforth.

### 4.1.3 Phonetic alignment

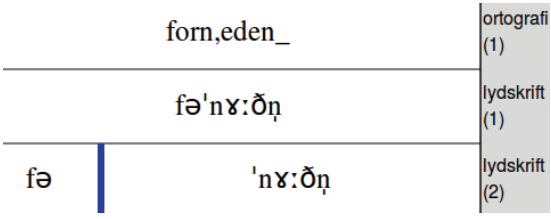
All phonetic annotations are mapped into IPA and represented in utf-8 character encoding and a boolean indicator variable is extracted from the phonetic annotation to indicate the presence of stød in a sample. Unlike the phonetic transcription in the JHP sample, stress is annotated as a diacritic on phones in DanPASS-mono and Parole48. To make the phonetic symbols as comparable as possible, stress annotation is removed from all phonetic annotations. DanPASS-mono and Parole48 also lack a phone level alignment as shown in Figure 4.1. Creating a phone level alignment manually was not feasible and the transcription in Parole48 and DanPASS-mono is automatically segmented.

In ASR, alignment between transcription and sound is computed using a two-step iterative machine learning algorithm. The method is known as embedded training using the Expectation-Maximisation algorithm and computes an alignment and a segmentation at the same time. In the first step, an equidistant segmentation of the speech data is assumed for each recording and aligned to the transcription symbols. From this alignment, a simple model for each annotation symbol is computed in the Expectation step and using these models, a new segmentation is computed in the Maximisation step and aligned to the transcription symbols. The algorithm continues until a fixed number of iterations have been computed or convergence is reached, i.e. there is little or no difference in the segmentation/alignment between each iteration.

Embedded training is scale-dependent and the data in this study (3 h 39 min. in total) cannot be considered large scale and the representation of individual labels is very small. Instead, a heuristic segmentation approach similar to the first step in embedded training was employed to segment and align the data. The heuristic approach applied here uses the following steps:




(a) Phonetic transcription of *mulighederne* from Parole48.




(b) Phonetic transcription of *forneden* from DanPASS-mono.

Figure 4.1: Phonetic transcription from DanPASS-mono and Parole48. The segmentation is above segment level and also sometimes above syllable level, e.g. [heD!C].

1. Divide a transcription $D$, e.g. [ˈkɛlɑːn̩] (DA: kælderen, EN: basement) , into $d_i$ phones, i.e. [ˈk ɛ l ɑː n̩] ($I = 5$)

2. Detect whether $d_i$ contains annotation for duration such as [ɑː]

3. Weight segments $d_1, d_2, ...d_I$ according to duration. $w(d_i) = 2$ for all phones unless the phone is suffixed with [ː] where $w(d_i) = 3$.

4. Divide the transcription duration $D^T$ by the sum of the segment weights: $\frac{D^T}{\sum_i^I w(d_i)} = d_i^t$. $2d_i^t$ is the duration of a segment, $w(d_i)$ is the duration weight of the segment at index $i$. $3d_i^t$ is the duration of long vowels e.g. [ɑː].

If the pronunciation of our example, [ˈk ɛ l ɑː n̩], takes 650 ms., the duration of a phone is estimated to be $2d_i^t = 118.18$ms and the duration of a long vowel is estimated to be $3d_i^t = 177.27$ms.

The heuristic relies on the existing time-coded transcription to extract the duration of words or syllables and uses the syllable and word boundaries to guide segmentation. The quality depends on the manual annotation of time-codes and the original annotation level, i.e. word-to-phone segmentation is likely to

be less accurate than syllable-to-phone segmentation. We have applied the heuristic alignment to map word-level and syllable-level alignment to a phonetic alignment.

### 4.1.4 Feature extraction

All features described here are extracted using short-term acoustic analysis (See Section 2.4). We use three different software toolkits – Praat, Covarep and Kaldi – to extract the features described in Chapter 2, because all features cannot be extracted using a single toolkit. We use a sample shift of 10 ms because the application scenario is ASR where 10 ms seems to be a de facto standard[2]. The size of the context window used depends on the feature.

Amplitude and harmonics-to-noise ratio are extracted using the `To Harmonicity (cc) ...` function in Praat. The function outputs one measurement for amplitude and an n-best list for harmonics-to-noise ratio measurements. The most likely hypothesis is chosen as harmonics-to-noise ratio for experimentation.

24 MFCC features, 8 glottal flow parameters and 38[3] phase features are extracted using Covarep. Covarep is a repository for speech analysis tools implemented in Matlab/Octave. Degottex et al. (2014) created the repository to share implementations of complex methods for speech analysis such as phase processing, glottal flow parametrisation and pitch tracking with other researchers and make it easier to reproduce research results.

39 PLP, 3 probability-of-voicing, 3 Pitch and 3 ΔPitch[4] features are extracted using Kaldi (Povey et al., 2011). The aims of the Kaldi project is similar in many respects to Covarep. The Kaldi Pitch Tracker (Ghahremani et al., 2014) implements a version of the Robust Algorithm for Pitch Tracking (Talkin, 1995)[5]. The main difference is that the algorithm does not make binary voicing decisions for a frame, but assigns a probability and in unvoiced regions, interpolate pitch values from adjacent frames in a straight line.

#### 4.1.4.1 Feature preprocessing

As a first step, audio is band-filtered through a low-pass Hann filter. The band filter removes frequencies above 1 kHz. The boundary was chosen manually by the author by listening to the DanPASS monologues, so stød is maximally audible while removing high frequencies. Then all features mentioned in Section 2.4 are extracted from the band-filtered audio.

[2] See the CMU Sphinx FAQ: `http://www.speech.cs.cmu.edu/sphinxman/FAQ.html`. All English Kaldi recipes also use 10 ms sampling shift.

[3] 25 PDM and 13 PDD measurements.

[4] First and second order derivatives are included.

[5] Before calculating pitch values, the Kaldi Pitch tracker also low-pass filters audio at 1 kHz.

Harmonics-to-noise ratio becomes undefined for non-harmonic regions of speech. Praat assigns a value of $-200$ to these regions which is problematic for estimation of means and variances, which is necessary for machine learning algorithms that assume a Gaussian distribution of the data. To alleviate the problem, we compute the minimum harmonics-to-noise ratio value $HNR_{min}$ on the harmonic regions of speech in the training data, i.e. where HNR$\neq$-200. $HNR_{lowbound}$ is then $HNR_{min}$ rounded down to the nearest 10. The equation is

$$HNR_{lowbound} = \left\lfloor \frac{HNR_{min}}{10} \right\rfloor * 10 \tag{4.1}$$

All samples with harmonics-to-noise ratio values of -200 are reset to $HNR_{lowbound}$. Test data is normalised using $HNR_{lowbound}$ calculated on training data. If resetting is not done, the subsequent scaling will be meaningless as the value -200 is arbitrary and chosen by the developers of Praat.

Subsequently, the acoustic features are standardised. Standardising features is a prerequisite for many machine learning classification methods such as GMM or SVM. A standard approach is to apply *mean subtraction* or *centering* followed by *feature scaling*. In the first step, we subtract a mean value calculated on the training data to 'center' the data. We then divide each feature by that features standard deviation to scale the variance across features. By standardising the parameter scales, the information contribution of a feature with a range of e.g. [-0.5, 0.5] (Peak Slope) is not overshadowed by a feature with a range of [0, 440] (Pitch).

The aim is to make the data resemble a Gaussian with zero mean and unit variance, because classifiers may not perform as expected unless data is properly standardised. However, mean and standard deviation can be computed on different basis. The most common examples are per utterance, recording session, speaker or corpus. In ASR, Cepstral Mean and Variance Normalisation subtracts a mean estimated per utterance and is designed to reduce channel noise whereas Vocal Tract Length Normalisation estimates a mean per speaker.

We experimented with speaker-, corpus- and gender-based means but did not observe any change in performance. A simple global cross-corpus feature standardisation is computed on the training set and applied to the acoustic features of both the training and test set.

## 4.2 Feature salience

Using all the extracted features from Section 2.4, each frame is represented by 120 features. Out of these features, some are known correlates of stød and some are not guaranteed to be salient for stød detection. Salient features have been discovered using mainly qualitative methods and a multi-speaker quantitative

analysis has not been conducted. Salient features can be discovered and the salience of existing correlates can be ranked using statistical analysis. The analysis results can also guide feature selection and elimination and reveal novel correlates that can suggest new directions where an answer to "...more than creak" can be found. It is also interesting to compare feature salience in training data to test data.

For dense features, 120 is a high dimensional space and it is unlikely that an ASR system will benefit from this many features. The limited training data and the high number of features require that steps be taken to avoid overfitting. Many of these features have been used to detect emotion or speech disorders (Degottex et al., 2014) and a model could learn to detect other facts about the training data. As the size of the feature vector increases, an ever higher number of samples are required to model the data accurately. This is the well-known effect of the *curse of dimensionality*. This is one of the primary reasons for including more data into the analysis and also a reason to investigate whether it is possible to reduce the feature space.

Reducing the number of features will give insight into the nature of stød by revealing the features which provide the most relevant information for stød detection. For a subset of information-rich features, it would be possible to create higher-order polynomial features for classification. Hard classification problems can sometimes be solved using polynomial and interaction features, e.g. if a feature set is not sufficient to separate the classes in the feature space. Especially, interaction features could give additional insight into the nature of stød.

### 4.2.1 Feature ranking

We perform a feature ranking experiment using the *extremely randomised trees* algorithm (extra-trees) which estimates an ensemble model or a *forest* of randomised decision trees. Each decision tree is fully grown top-down by node splitting and before each split, a random subset $F_r$ of all features $F$ is chosen and for each feature $f_i$ in $F_r$, a cut-point is chosen at random. The feature $f_i$ with cut-point $a_i$ which most improves entropy after the split is used to split the data in the node. The reduction in entropy is calculated as the entropy at the parent node $p$ subtracted entropy at the child nodes ($n_{left}$ and $n_{right}$) and entropy is weighted by sample size at the nodes ($n_{node}$):

$$reduction_{f,p} = entropy_p * n_p - entropy_{left} * n_{left} - entropy_{right} * n_{right} \tag{4.2}$$

In concrete terms, feature importance is the total reduction in entropy provided by a feature. Because a feature can be used multiple times with different cut-points in a decision tree, we sum the relative entropy reduction ($\sum_p^{P_f} reduction_{f,p}$) of a feature $f$ across all parent nodes that branch on feature $f$ ($P_f$)

in the tree before normalising by the total number of samples ($N$) and by the sum of feature importances ($\sum_f^F Importance_f$):

$$ImportanceNorm_f = \frac{1}{\sum_f^F Importance_f} * \frac{\sum_p^{P_f} Reduction_{f,p}}{N} \tag{4.3}$$

Decision trees are notoriously unstable because a different tree can be constructed to achieve the same classification. If the structure is different, the feature ranking will change. By estimating a large forest of trees where branching decisions are selected based on random cut-points and random feature subsets, we can compute robust estimates of feature importances by averaging over trees in the forest, because the effect of irrelevant features are reduced.

In other words, a feature can be ranked by the relative depth at which it is used to split a node in a decision tree. The features used at nodes close to the root node of a decision tree are more discriminative than features used near the leaf nodes of a tree. The motivation is that features used at the top of a tree contribute information that can discriminate more samples in classes. The fraction of samples, that a feature contributes to discriminate, is averaged over many trees and is used to rank features by relative importance (Geurts et al., 2006).

### 4.2.2 Experiment setup

Using the extra-trees algorithm, we train a forest of decision trees on various subsets of the training data. Bootstrapping, or sub-sampling with replacement, ensures that the same data can be used to train decision trees which is necessary due to the under-representation of stød-bearing samples in the data. With limited training data available, bootstrapping provides more robust estimates than reducing the samples in each class will afford.

Each forest consists of 1024 decision trees, the classes have an equal weight and entropy is used as the criterion to measure the quality of splitting a node. The minimum number of samples required to split a node is 2 and there must be at least one sample in a leaf node.

Because node splitting is scale invariant, features are not preprocessed as in Section 4.1.4.1 and non-normalised data is used for feature ranking. However, the amount of samples in each class is held equal, so important features for stød detection are not drowned by features that classify non-stød classes, which is over-represented in the data. Decision trees were chosen over parametric methods such as L1 regularisation-based feature selection because Gaussian distribution of the data is not assumed. The motivation is to make the feature ranking independent of implementation differences between Praat, Covarep and Kaldi and standardisation.

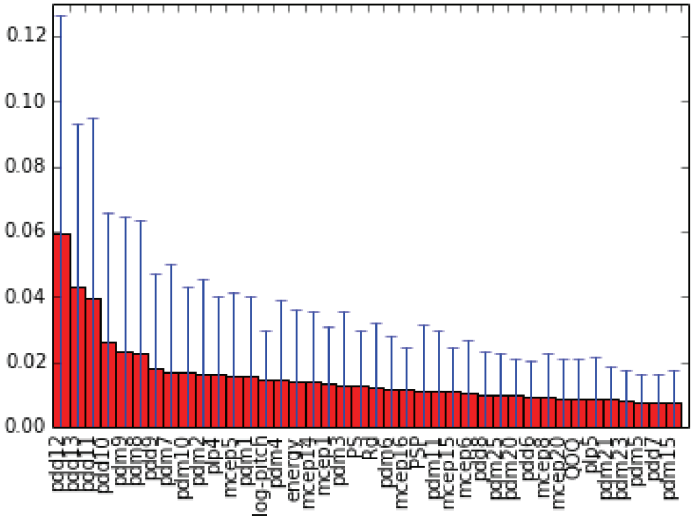



Figure 4.2: Feature importance for discriminating between [ɑːʔ] vs. [ɑː]: Train data.

Feature ranking reveals that out of 120 features, 116 are salient for stød discrimination in the training data. The features that are not important for discrimination are PDD features 1-4 in the training data and PDD 5-6 in the test data.

### 4.2.3 Stød-bearing vs. stød-less

A comparison of feature salience for discrimination between the phones [ɑːʔ] vs. [ɑː] can be seen in Figures 4.2 and 4.3.

In Figure 4.2, PDD and PDM occur frequently among the top 40 features. Pitch, energy, Peak Slope (PS), Rd, quasi-open quotient (QOQ) and the parabolic spectral parameter (PSP) are also present in the figure. Pitch and Rd are salient in the test data as well as the generally prevalent phase distortion means (PDM). Also raw-log-pitch-delta and harmonics-to-noise ratio (HNR) are salient according to Figure 4.3.

Finally, a two-part ranking experiment that takes into account all annotations with stød and their stød-less counter-parts have been conducted. The distribution of stød is unequal between training and test data as well as within the data sets. The raw frequency can be seen in Appendix A.1.

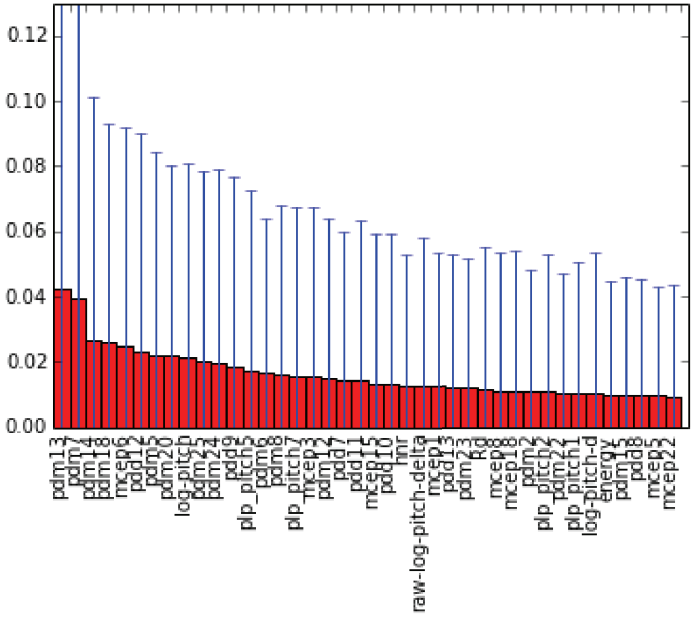



Figure 4.3: Feature importance for discriminating between [ɑːˀ] vs. [ɑː]: Test data.

Instead of filtering globally such that the amount of stød-bearing samples and stød-less samples have the same size, the individual classes in each data set are filtered to have an equal number of samples in each class. If no samples for a class are present in one of the data sets e.g. as is the case for ðʏˀ, all ðʏˀ and ðʏ samples in the data are excluded from the analysis.

All remaining phones were used to create the forest of decision trees. The resulting feature rankings can be seen in Figures 4.4 and 4.5. The second part of the experiment entailed converting the phones into a boolean value of +/- stød and repeating the experiment. The feature rankings can be seen in Figures 4.6 and 4.7.

To discriminate between stød-bearing and stød-less samples, Peak Slope, probability-of-voicing, energy and Δenergy, intensity and quasi-open quotient are salient in Figure 4.4. Phase-related features also figure in the top 40, but do not dominate as previously observed.

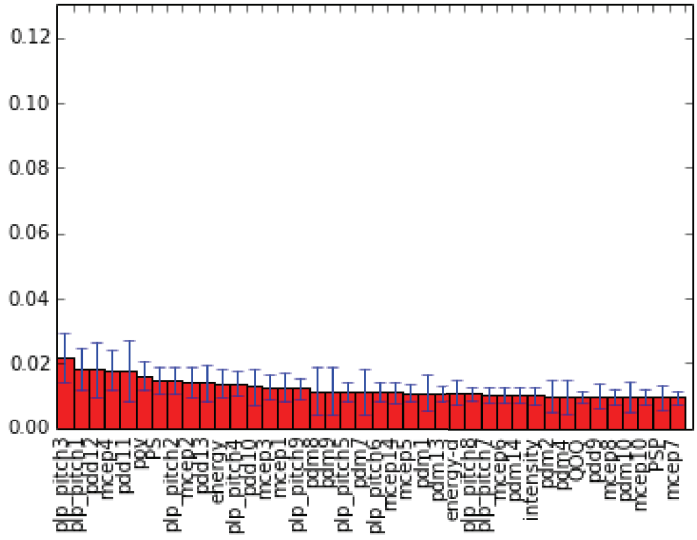



Figure 4.4: Feature importance for discriminating between stød-bearing and stød-less samples in train data.

To discriminate phones in the test data, phase features, pitch, and Peak Slope are salient again, but log-pitch figures as important in Figure 4.5 in contrast to 4.4.

To detect stød in training data across segments, i.e. in the binary case, probability-of-voicing, energy and pitch features with Δ-derivatives, Peak Slope, normalised amplitude quotient, quasi-open quotient, intensity, harmonics-to-noise ratio and H1-H2 are relevant features.

This is a very striking difference to the test data where phase features figure frequently as the most salient features in addition to Peak Slope, energy and pitch information.

### 4.2.4 Analysis of feature ranking

The difference in salient features between training and test data indicate that feature selection may be appropriate for detection experiments. Consistently, phase features, Peak Slope and to a lesser extent HNR, probability-of-voicing and pitch-related features have figured among the most salient features. As expected, the ASR-related MFCC and PLP features also add discriminative information, but their Δ and ΔΔ-derivatives consistently figure as the least informative features.

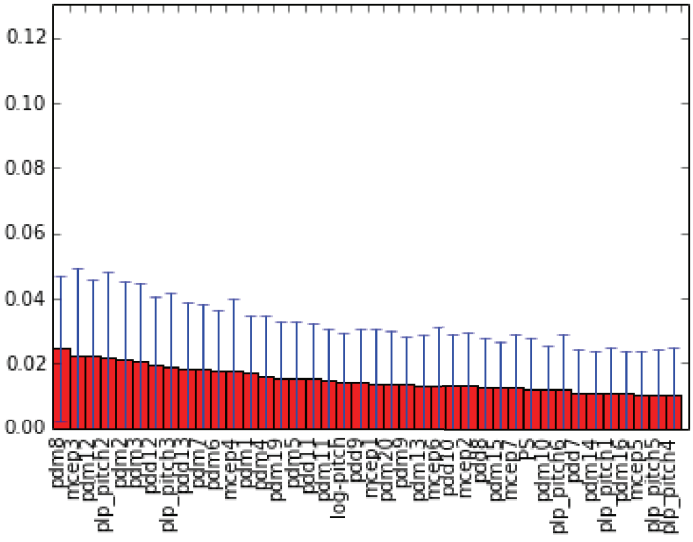



Figure 4.5: Feature importance for discriminating between stød-bearing and stød-less samples in test data.

That phase information is salient for stød detection is to my knowledge a novel insight. It is an interesting discovery because PDM and PDD rank higher than many ASR-related features such as ΔPLP and ΔΔPLP-derivatives and MFCC features. If this finding can be corroborated in the analysis of other corpora, phase features might be useful information to add to acoustic models in ASR.

A common set of features that are salient for phone discrimination and stød detection emerges from studying Figures 4.4 and 4.6. The X-axes suggest to select PDM features 1,2,4,7,8,9,10,13 and 14, PDD 10-13, MFCC and PLP features 1-6, Peak Slope, Rd and pitch for stød detection. The top 10 features would be PLP 1-4, PDD 10-13, Peak Slope, probability-of-voicing and log-pitch.

If the classic ASR features MFCC and PLP are not included, the top 10 most salient features are PDD 10-13, PDM 13-14, Peak Slope, log-pitch, probability-of-voicing and Δprobability-of-voicing. The most salient ASR features are PLP and MFCC 1-4. Thus, the ranking experiment indicates that stød detection in ASR could benefit from adding the following features:

1. MFCC 1-4 if the system is based on PLP features and vice versa

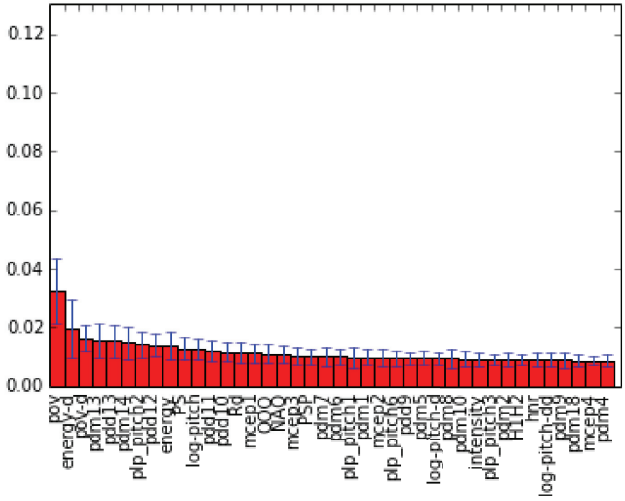



Figure 4.6: Feature importance for binary classification of stød in train data.

2. Peak slope, log-pitch, probability-of-voicing and Δprobability-of-voicing

3. PDD 10-13, PDM 13-14

Subsequently, the feature sets above will be referred to as speech recognition feature set, glottal feature set and phase feature set. The set comprised by the combination of these features will be denoted *select+*.

Energy also ranks highly in the ranking experiments, but has not been selected as a salient feature. The energy, Δenergy and ΔΔenergy features are part of both PLP and MFCC features where they replace the zeroth coefficient and as such included in all ASR experiments by default.

The feature ranking does not take into account feature interaction. The high dimensional feature space made it impossible to use polynomial features in the ranking experiments due to hardware, software and time limitations. Calculating polynomial features on select+ only did not alleviate the limitations.

## 4.3 Detection experiments

The feature salience analysis Section 4.2.4 indicates a subset of features that are correlated with stød. In this section, classifiers trained on select+ features are compared to classifiers trained on all 120 features.

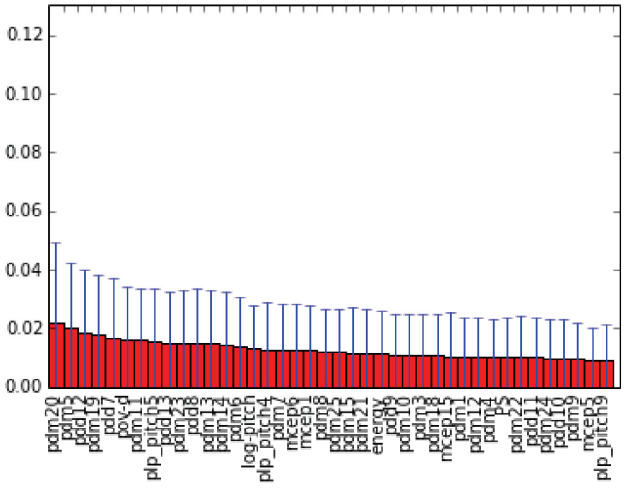



Figure 4.7: Feature importance for binary classification of stød in test data.

Additionally, PCA has been applied to project the full data set into 40-dimensional feature vectors which will also be reported on. The motivation is that select+ includes the most salient features, but a large number of features have been discarded including features that have previously been reported to be salient for stød. Feature projection may be able to retain more information and outperform feature selection.

The problem of stød detection can be formulated in two ways: a binary and a multi-class classification task.

The binary classification task is the most intuitive formulation. Classifiers are trained in an effort to learn statistical models that predict the occurrence of stød. The classifier could add a binary or probabilistic feature to the input vector of a downstream NLP application. If the downstream application is ASR, the feature can inform discrimination between stød-bearing and stød-less sonorants.

In the multi-class formulation of stød detection, models that can discriminate between stød-bearing phones and their stød-less counterparts are estimated. Instead of creating a classifier that classifies a sample as stød-bearing or stød-less, we try to predict several stød classes, i.e. stød-bearing phones which is analogous to mixing up in ASR. The model will not be able to untangle which features are used to detect

stød and which are salient for a phone, but this discrimination is what stød detection would be useful for in downstream NLP applications such as ASR.

### 4.3.1 Annotation transformation

According to stødbasis, stød is annotated on

1. A long vowel
2. The subsequent sonorant if the vowel is short

The stød detection experiments are repeated on data where stød annotation is extended to include the short vowel in the case of 2.

### 4.3.2 Classifiers

NB classifiers provide good classification performance if the training data is biased. Large-margin classifiers such as SVMs usually perform well on large collections of data and can find a better decision boundary than NB classifiers. Scikit-learn wraps `libsvm` (Chang, Chih-Chung and Lin, Chih-Jen, 2011) and `liblinear` (Fan et al., 2008) for training large margin classifiers. Also Logistic Regression (LogReg) and GMM classifiers will be evaluated. Because of the high dimensionality of the data, Stochastic Gradient Descent optimisation is used to minimise the objective function for LogReg and SVM.

The parameters of each algorithm are found by exhaustive grid search on a specified parameter space and 10-fold cross validation on 90% of the training data and evaluated according to F1 on 10% held-out data. The classifiers are sensitive to the regularisation parameter $\alpha$ and values ranging from 10 to 0.00001 have been searched.

Using the best performing SVM and LogReg classifiers, different sample weights will be used to re-estimate the classification boundary and maximise F1. If the samples for the stød-bearing class(es) in the data are noisy, a reduced sample weight for that class could improve the decision boundary by reducing the amount of false positives and noise could have been introduced by both the manual annotation and the heuristic alignment. Vice versa, if the number of false negatives is high, a higher sample weight for the stød-bearing class could improve classification. Setting sample weight for all classes to 1 is equal to no sample weight.

The GMM classifier was not trained using cross validation. Instead, grid search over the number of expectation-maximisation iterations and covariance types[6] was performed. GMM parameters (weights, means and covariances) are initialised 50 times, and the best initial parameter set is kept.

[6]Spherical, tied or diagonal covariances.

### 4.3.3 Binary classification experiment

In the binary stød detection experiments, performance of the classifiers will be evaluated on the development set according to recall, precision and F1-score. Precision is measured as

$$Precision = \frac{True\,positives}{True\,positives + False\,positives} \tag{4.4}$$

For a perfect classifier, precision (and recall) becomes 1. As the proportion of False positives increases, precision decreases. In the context of the experiments in this chapter, precision can be described as the ability of a classifier to not classify samples that are stød-less as stød-bearing.

Recall is measured as

$$Recall = \frac{True\,positives}{True\,positives + False\,negatives} \tag{4.5}$$

Recall will decrease as the proportion of False negatives increase. Recall is interpreted as the ability of a classifier to label all samples in the data annotated with stød as stød-bearing.

F1 is the harmonic mean of precision and recall where recall and precision both have equal weight. The parameter set and sample weight that optimises F1 on the development set is used to train the classifier on all the training data before evaluating on the test data.

#### Results

To distinguish the different data sets used, experiments using the original data set will be referred to as *raw* while experiments using extended annotation will be denoted as *extended*. In addition, the feature set names *full*, *select+* and *PCA* will indicate the acoustic features used in the experiment.

The classification results on development data with classifiers trained on the full features set is displayed in Table 4.1.

Grid search finds regularisation values from 0.001 and smaller (SVM and LogReg.) indicating that the best performance is obtained with sparse statistical models. This is an indication that some parameters are superfluous for the classification task, i.e. the coefficients for some features become zero and does not inform the classification.

F1 is maximised using a sample weight of 0.3 for the stød-bearing class in the raw condition. When the annotation is extended, we see an increase in F1 for all classifiers with the exception of GMM. The optimal sample weight found in the extended condition for both classifiers is 0.4, which is higher than the sample weight found in the raw condition.

| | Raw | | | Extended | | |
|---|---|---|---|---|---|---|
| Classifier | Precision | Recall | F1 | Precision | Recall | F1 |
| NB | 0.10 | 0.69 | 0.18 | 0.14 | 0.69 | 0.24 |
| GMM | 0.08 | **0.94** | 0.14 | 0.02 | 0.09 | 0.03 |
| LogReg | 0.13 | 0.79 | 0.22 | 0.17 | 0.76 | 0.28 |
| LogReg+sw | 0.24 | 0.40 | 0.30 | **0.22** | 0.61 | **0.32** |
| SVM | 0.13 | 0.80 | 0.22 | 0.17 | **0.78** | 0.28 |
| SVM+sw | **0.27** | 0.36 | **0.31** | 0.21 | 0.60 | **0.32** |

Table 4.1: Precision, Recall and F1 for classifiers trained on the full feature set. The best metrics in a column is bold faced. LogReg and SVM was refit using sample weight 0.3/1 in the raw condition and 0.4/1 in the extended condition for stød-bearing/stød-less classes, respectively.

#### 4.3.3.1 Classification with feature selection

The experiments are repeated using the *select+* feature set and the results are reported in Table 4.2. Apart from the feature set, the experimental setup is identical to the experiments in the section above.

| | Raw | | | Extended | | |
|---|---|---|---|---|---|---|
| Classifier | Precision | Recall | F1 | Precision | Recall | F1 |
| NB | 0.12 | 0.31 | 0.17 | 0.16 | 0.32 | 0.21 |
| GMM | 0.12 | **0.76** | 0.21 | 0.12 | **0.76** | 0.21 |
| LogReg | 0.14 | 0.73 | 0.23 | 0.15 | 0.68 | 0.24 |
| LogReg+sw | **0.18** | 0.50 | **0.27** | 0.19 | 0.43 | 0.26 |
| SVM | 0.14 | 0.74 | 0.23 | 0.14 | 0.74 | 0.23 |
| SVM+sw | **0.18** | 0.51 | **0.27** | **0.20** | 0.39 | **0.27** |

Table 4.2: Precision, Recall and F1 for classifiers trained on select+ features. The best metrics in a column is bold faced. LogReg and SVM was refit using sample weight 0.4/1 for stød-bearing/stød-less classes in the raw condition and 0.5/1 in the extended condition.

Comparing the raw/select+ condition in Table 4.2 to the raw/full condition in Table 4.1, the GMM classifier performs better by 0.07 F1 absolute and NB, where recall is less than half of the previous experiment, performs worse by 0.01 F1. F1 scores are higher using select+ features for LogReg and SVM, but only by

0.01 F1 absolute and we see that an increase in precision counterbalances a lower recall. Sample weighting reverses the comparison and the raw/full condition obtains higher F1. The effect of sample weighting is to even out the imbalance between precision and recall but neither LogReg. nor SVM outperforms the classifiers in Table 4.1.

Using the extended annotation, the NB classifier performance improves by 0.04 F1 absolute compared to the raw condition in Table 4.2. GMM results are not influenced by the change in annotation and the SVM classifier performs identically. As the only classifier, GMM performs better using select+ features in both annotation conditions. Sample weighting improves F1 scores, but the classifiers do not reach the same level of performance using select+ features.

#### 4.3.3.2 Classification with feature projection

The performance of classifiers trained on the full feature set outperforms classifiers trained on select+ features. Linear dimensionality reduction using PCA may be able to reduce the number of features and retain more information than feature selection. Due to hardware limitations, batch PCA could not be applied to the full feature set and Incremental PCA is applied to learn a model to project the full feature vectors to 40 dimensions. The results can be seen in Table 4.3.

| | Train | | | Development | | |
|---|---|---|---|---|---|---|
| Classifier | Precision | Recall | F1 | Precision | Recall | F1 |
| NB | 0.14 | 0.31 | **0.19** | 0.14 | 0.31 | **0.19** |
| GMM | 0.03 | 0.24 | 0.06 | 0.03 | 0.24 | 0.06 |
| LogReg. | 0.10 | 0.74 | 0.18 | 0.10 | 0.74 | 0.18 |
| SVM | 0.10 | 0.76 | 0.18 | 0.10 | 0.77 | 0.18 |

Table 4.3: Precision, Recall and F1 for classifiers trained on 40-dimensional PCA projected training data.

This experiment only includes a raw/PCA condition. Using incremental PCA for stød detection reduces precision compared to the select+ conditions, but does not lead to an increase in recall and results in comparatively low F1 scores on both training and development data. Incremental PCA projection to 60, 80 and 100 features have also been performed to determine whether a 40-dimensional feature space was too small to retain salient information, but PCA projection does not outperform feature selection.

**Exponential features**

1st, 2nd and 3rd order exponential features were computed on select+. The best NB classifier obtained 0.18 F1 on both training and development data and the best SVM achieved 0.22 F1. Like incremental PCA projection, exponential and interaction features did not improve performance over classifiers trained on full or select+ features.

#### 4.3.3.3 JHP evaluation

In this section, classifiers trained on raw/full, raw/select+, extended/full and extended/select+ conditions are evaluated on the JHP data set. The annotation created by IPA3 will form the basis for evaluation. IPA3 was chosen because the MACE evaluation ranked IPA3 as one of the two most competent annotators.

The evaluation of classifiers trained on raw annotations can be seen in Table 4.4. Compared to training and development set evaluation, a decrease in all measures are observed in Table 4.4. The spontaneous speech genre poses a difficult task for the classifiers trained on data extracted from elicited speech. Using select+ features leads to generally higher precision while classifiers trained on full feature set favours recall. It is however possible to achieve the same F1 score using both feature sets.

The effect of sample weights is a large decrease in F1 in the JHP evaluation. While sample weighted classifiers achieve the best precision, recall is between 0.13 and 0.02.

| | Full | | | Select+ | | |
|---|---|---|---|---|---|---|
| Classifier | Precision | Recall | F1 | Precision | Recall | F1 |
| NB | 0.09 | 0.28 | 0.13 | 0.08 | 0.08 | 0.08 |
| LogReg | 0.11 | 0.45 | **0.17** | 0.12 | 0.29 | **0.17** |
| LogReg+sw (0.3/0.4) | 0.18 | 0.08 | 0.11 | **0.33** | 0.02 | 0.03 |
| SVM | 0.10 | **0.52** | 0.16 | 0.12 | 0.33 | **0.17** |
| SVM+sw (0.3/0.4) | **0.19** | 0.13 | 0.15 | 0.24 | 0.04 | 0.07 |
| GMM | 0.03 | 0.28 | 0.05 | 0.07 | **0.73** | 0.13 |

Table 4.4: Precision, Recall and F1 evaluation for binary classification on the JHP sample. The best metrics in a column is bold face.

In the test condition with extended annotation, sample weighting leads to worse performance than using equal sample weighting (No sample weights). SVM achieves a higher F1 score than other classifiers using both select+ and full feature sets in Table 4.5. While extending the annotation has a tendency to decrease

recall on training and development data, a general increase in recall is observed for the full feature set and a slight reduction in precision. The same influence on performance for extended annotation can be seen in the select+ conditions.

| | Full | | | Select+ | | |
|---|---|---|---|---|---|---|
| Classifier | Precision | Recall | F1 | Precision | Recall | F1 |
| NB | 0.08 | 0.31 | 0.13 | 0.08 | 0.08 | 0.08 |
| LogReg | 0.10 | 0.59 | **0.17** | 0.09 | 0.31 | 0.14 |
| LogReg (0.3/0.5) | **0.16** | 0.09 | 0.11 | 0.18 | 0.02 | 0.03 |
| SVM | 0.10 | 0.65 | **0.17** | 0.10 | 0.32 | **0.15** |
| SVM (0.3/0.5) | 0.14 | 0.13 | 0.13 | **0.19** | 0.09 | 0.12 |
| GMM | 0.07 | **0.72** | 0.13 | 0.07 | **0.73** | 0.13 |

Table 4.5: Precision, Recall and F1 evaluation for binary classification on the JHP sample with extended annotation. The highest F1 scores are in bold face.

The effect of sample weight is stable across annotation and feature set conditions. Sample weights optimised according to F1 on development data have an adverse effect on evaluation on JHP data.

### 4.3.4 Discrimination experiment

A slightly modified version of the experiment in Yoon et al. (2006) will be applied to classifying stød-bearing and stød-less samples.

This experiment will also use a SVM classifier, but with a radial basis function kernel, to classify stød-bearing and stød-less samples. Like Yoon et al., the same underlying implementation is used and the parameters of the SVM are left unchanged and not optimised. To evaluate performance, classification accuracy with three feature sets will be reported:

**PLP** PLP features extracted with Kaldi

**All** All features extracted with Covarep, Kaldi and Praat

**Select+** The feature set used above with 17 salient features.

Following the methodology in the existing study, PLP features will be used as a baseline. Classification accuracy using all and select+ features will give indication whether the difference between stød-bearing and stød-less segments is reflected to a greater or lesser extent in features not commonly used in ASR.

Because SVMs with a radial basis function kernel scale badly to large amounts of data, the evaluation follows a One-vs-One setup rather than the classical One-vs-All used in binary stød detection. One-vs-One evaluation is similar to the ranking experiment in Section 4.2.4. The evaluation is considered relevant because this experiment seeks to discover whether the input features contain sufficient information to discriminate stød-bearing and stød-less variants of the same phone.

Five-fold cross validation scores will be reported instead of performance on 10% held out data. Five folds were chosen because some segments have very low representation in the training data (n=9 samples). In addition, the performance on JHP data will be reported for those annotations that are in common between training and test data.

**Annotation granularity**

Yoon et al. (2006) achieve an overall classification accuracy of 69.23% on 25 minimal phone pairsin One-vs-One evaluation and indicates that voice quality is to some extent reflected in PLP features. In the training and test data, annotation uses semi-fine IPA and distinction is made between long and short vowels, syllabification and creak, which results in 40 phone pairs.

This is not a distinction that is usually observed in ASR, because temporal variation is modelled with recursive transitions in HMMs, and we therefore map to a more coarse phonetic alphabet such that e.g. [m̥]→[m], [aː$^{ʔ}$]→[a$^{ʔ}$] and [v̰]→[v], which creates conditions more similar to the previous study with 29 phone pairs.

#### 4.3.4.1 Results

A summary of the One-vs-One five-fold cross validation results can be seen in Table 4.6. A detailed output of the evaluation can be seen in Appendix A.2 and A.3. Similar to Yoon et al. (2006), the results of the evaluation show that PLP features contain information pertinent to distinguish between stød-bearing and stød-less sonorants. In both annotation granularities, the accuracy is very high – especially compared to the experiments in the previous chapter.

The classification accuracy improves with voice quality features and select+ achieves the highest accuracy followed by the full feature set with semi-fine annotation. In the evaluation, the variance increases together with accuracy and the difference cannot be considered significant as the mean accuracies are within the variance of classifiers trained on other feature sets.

| | Full | | PLP | | Select+ | |
|---|---|---|---|---|---|---|
| Alphabet | Accuracy | +/- | Accuracy | +/- | Accuracy | +/- |
| Semi-fine IPA | 0.781 | 0.168 | 0.769 | 0.144 | **0.803** | 0.176 |
| Coarse IPA | **0.922** | 0.058 | 0.885 | 0.096 | 0.853 | 0.119 |

Table 4.6: Summary table for Five-fold One-vs-One evaluation on training data using different feature sets and alphabets. The full tables can be seen in Appendix A.The best performance for each alphabet granularity is in bold.

#### 4.3.4.2 Coarse phone discrimination

Using coarse IPA[7], the evaluation is slightly different. The classifiers trained in the full feature set now shows a near-perfect classification accuracy and a very low variance. The difference in accuracy is significant over the results of the evaluation of PLP and select+ feature sets. PLP is slightly more accurate than select+, but not significantly.

Looking at the detailed output in Appendix A, classes with low representation, such as [aːˀ]/[aː], [ɐ̯ˀ]/[ɐ̯] and [m̩ˀ]/[m̩], are merged into larger similar groups. The classes with small representations were classified with high accuracy by the SVM trained on select+ features, while the classifiers trained on full and PLP sets had low accuracy and high variance. The absence of small classes and the increased sample size for large classes increase the mean accuracy for classifiers trained on full and PLP feature sets. Mean accuracy for select+ does not improve as much due to the absence of these classes.

#### 4.3.4.3 Evaluation on the JHP sample

Applying the trained classifiers to test data shows that discriminating stød-bearing and stød-less segments is more challenging. In Table 4.7, the mean accuracy is slightly better than chance, but the variance is quite high. The increased variance we observe for the full feature set and PLP features on the JHP sample indicate that they overfit to the training data. While select+ features do not obtain the same mean classification accuracy, the variance is similar and does not indicate overfitting.

[7]The term *coarse IPA* denotes that the symbol set used is smaller than fine IPA, not that the annotation follows the guidelines for coarse IPA presented in Grønnum (2005).

| Phones | | Samples | Full | PLP | Select+ |
|---|---|---|---|---|---|
| lˀ | l | 26 | 0.788 | 0.788 | 0.588 |
| mˀ | m | 5 | 0.700 | 0.900 | 0.600 |
| nˀ | n | 58 | 0.638 | 0.664 | 0.569 |
| ŋˀ | ŋ | 7 | 0.500 | 0.714 | 0.571 |
| ɒˀ | ɒ | 5 | 0.800 | 0.500 | 0.700 |
| Mean accuracy | | | 0.685 | **0.713** | 0.600 |
| Std.dev. | | | 0.220 | 0.266 | **0.104** |

Table 4.7: Stød occurrence and mean classification accuracy on the JHP sample for three feature sets.

### 4.3.5 Analysis

#### 4.3.5.1 Annotation

Conducting the same experiment with raw and extended annotation shows that using extended annotation gives higher precision and F1 scores than using the raw annotation. This is especially the case with NB classifiers. Looking at SVM, LogReg. and NB classifiers, recall remains stable while precision increases indicating that the proportion of false positives decrease or true positives increase. The improved evaluation suggests that some samples from the preceding vowel are in fact stød-bearing, but not annotated as such. Extending the annotation to include the preceding vowel of a stød-bearing sonorant is counter-acting the annotation convention related to stødbasis explained in Section 2.2.1.

Sample weights improve the performance of both SVM and LogReg. on training and development data. It is interesting that the sample weight for the positive class (stød-bearing class) is smaller than the negative class. To optimise classification, the decision boundary is less influenced by training samples from the positive class. That is an indication that the class is noisy and that there are samples in the positive class that have been wrongly labelled. The annotation error could have occurred during alignment or manual annotation.

#### 4.3.5.2 Features

120 is a high number of dense features. Dimensionality reduction using linear PCA was performed as a training preprocessing step. The motivation is that feature projection could preserve some of the discriminative information contained in features not included in select+ and that a reduced feature set could reduce the time spent estimating statistical models by learning a simpler model. The drawback is of course the

interpretability of the projected features. Training and development set evaluation shows that the performance of classifiers trained on the 40-dimensional projected features are not as good as those trained using simple feature selection with 17 features. Using select+ features, an improvement can be observed only for the NB classifier while the remaining classifiers perform similarly according to F1, yet with different precision and recall scores.

The high number of features is also due to the (almost) zero-knowledge approach to stød detection. That a number of features are irrelevant to stød detection is therefore not surprising, e.g. in the case of ΔPLP, ΔΔPLP, ΔMFCC and ΔΔMFCC because these features are engineered to model the speed and acceleration of speech organs that are not correlated with the glottis.

There is also the possibility of collinear features. Both non-salience and collinearity are supported by the regularisation parameter $\alpha$ for LogReg. and SVM classifiers which are constantly below 1.0. For $\alpha < 1.0$, the classifiers learn a sparse model. In sparse models, a number of learned coefficients become zero, yielding classification that relies on information from only a subset of features.

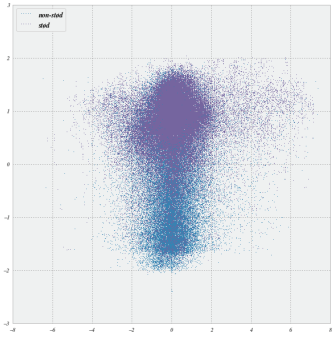
Though feature selection outperforms PCA, an average decrease in performance of 12.5% relative F1 (0.04 points absolute) compared to the full feature set is observed for classifiers trained on select+ features. Considering that 85.8% of the full feature set is discarded and similar performance con be obtained on the JHP sample, select+ retains most of the salient information for stød detection. Unweighted LogReg. and SVM classifiers achieve the same F1 score (with different precision/recall balance) using both full and select+ feature sets.

Although salient features have been discovered, the classes are not adequately separated in the feature space for linear classification. This is especially clear when some of the most salient features are plotted against each other in Figure 4.8.

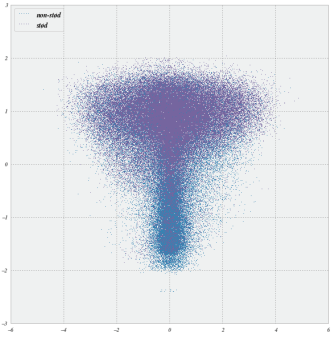
While some samples are separated from each other and could be classified using a linear classifier, there are many samples from both classes that are clustered together. The optimal sample weight found using F1 optimisation has consistently been lower than 1 for the stød-bearing class. This indicates noisy annotation and is supported by the sample weight found when re-estimating a decision boundary using logistic regression or SVM and by the scatter plots in Figure 4.8.

#### 4.3.5.3 Class skewness

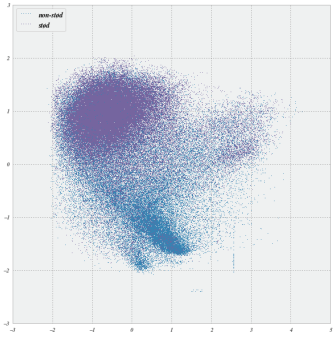
Due to the skewness of the distribution of stød, stød detection is a difficult classification task. Accounting for skewness using an inverse prior improves the precision/recall balance and optimising F1 rather than



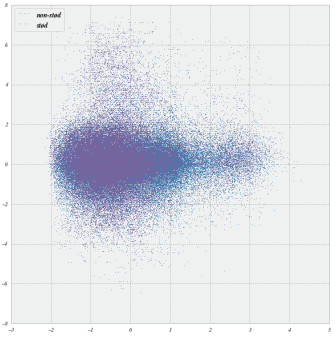

(a) Pitch vs. energy



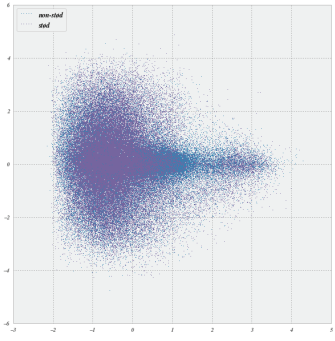

(b) Δprobability-of-voicing vs. energy



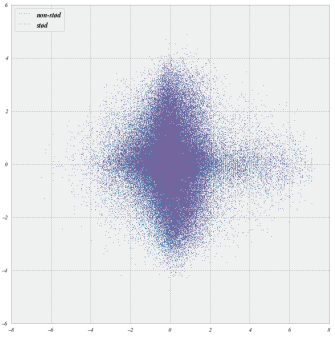

(c) Peak slope vs. energy




(d) Peak slope vs. Pitch




(e) Peak slope vs. Δprobability-of-voicing

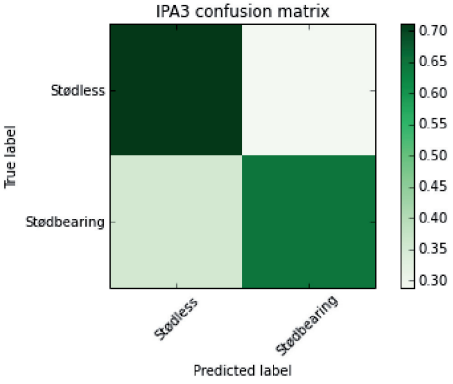



(f) Pitch vs. Δprobability-of-voicing

Figure 4.8: Training samples plotted by salient features according to feature selection. Stød-less samples are blue and stød-bearing samples are purple.

| | n | p |
|---|---|---|
| n | 6685 | 2707 |
| p | 163 | 297 |

(a) Unnormalised detection counts for stød detection.




(b) Normalised confusion matrix for stød detection.

Figure 4.9: Raw classification counts and confusion matrix normalised by class support for visual presentation. The counts in 4.9a correspond to classes in 4.9b. Classified using unweighed linear SVM in the extended annotation condition on JHP.

accuracy prevents the classifiers from learning a decision boundary that simply classifies all samples as stødless[8]. While the results are insufficient for practical application, the success of the classification is difficult to determine because of the skewness.

A classifier with a low number of false positives could still be useful for downstream applications in both academia and industry, i.e. a high precision classifier with low recall. The best performing classifier on JHP data is the unweighted SVM trained in the raw/select+ condition. The success can be visualised using confusion matrices. The raw development set classification counts and normalised confusion matrix can be seen in Figure 4.9.

The matrix in Figure 4.9a shows the counts of true negatives, false positives[9], false negatives and true positives from top left to bottom right. Figure 4.9b illustrates the true negative rate, false positive rate, false negative rate and true positive rate in the same order. Darker greens illustrate a higher rate after normalisation by the number of true class samples.

In this case, the true positive rate is high and the false positive rate is low which is desired in a high precision/low recall classifier. However, the proportion of false positives out of the total predicted positives

[8]Results in ca. 95% accuracy.
[9]Aka. false alarms.

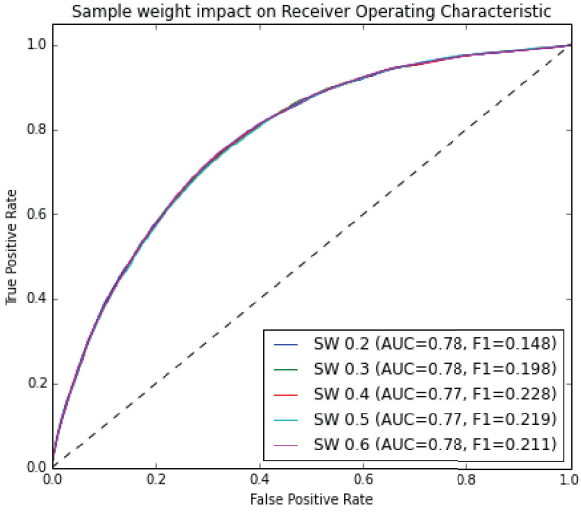



Figure 4.10: Reciever operating characteristic curves for different sample weights on development data.

(false discovery rate) is 0.9 and indicate that due to the low prevalence of stød, even the best classifier in our experiments is not able to learn a good decision boundary.

The classifiers do exhibit some desirable properties. This is best illustrated using Receiver Operating Characteristic curves. Figures 4.10 and 4.11 show graphs that plots true positive rate as a function of false positive rate. The dashed line corresponds to random classification. This plot illustrates that the classifiers predict stød better than chance on both development and test data.

#### 4.3.5.4 Discrimination experiment

Unlike stød detection, phone discrimination experiments shows a high degree of accuracy. Like Yoon et al. (2006), PLP features can to a certain degree discriminate between the stød-bearing and stød-less variants of the same phone. Those features can be replaced by full and select+ features to obtain similar results.

If some distinctions maintained in semi-fine IPA are removed, the evaluation improves significantly which indicates that the features included in select+ contain information that model distinctions that are not directly related to the stød+phone discrimination task. Removing these distinctions also adds a significant amount of training data, e.g. in the case of [ɔːˀ] which increases the number of training samples for [ɔˀ] by 1/3. Especially the PLP-based SVM gain significantly from the larger sample sizes.

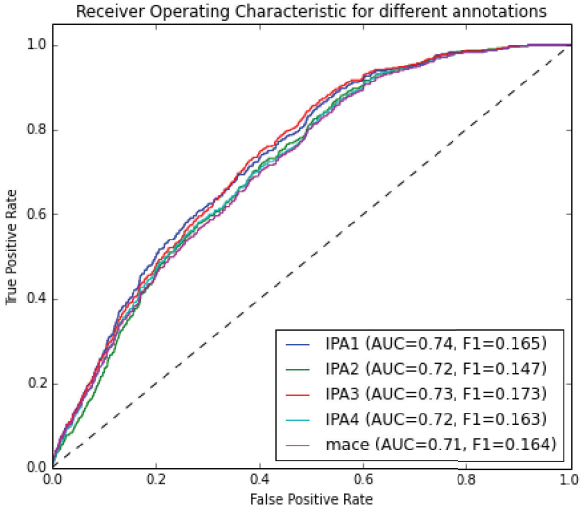



Figure 4.11: Reciever operating characteristic curves for all annotations on test data.

On the JHP sample, mean accuracy decreases while variance increases. The low variance in cross validation indicates that the statistical model overfits the training data and is not able to generalise to unseen data from a different speech genre. This is corroborated by the small variance for the classes from Table 4.7 in Appendix A.3 and the significant increase in variance on test data. Similar effects were observed in F1 evaluation in the binary classification experiments and feature ranking experiments.

The variance does not increase in the evaluation of the SVM trained on select+ features and while the mean accuracy is lower, the model did not overfit to the training data and therefore perform similarly.

## 4.4 Discussion

### 4.4.1 Features

As the feature selection experiments demonstrated, not all features are important for stød detection. Using a smaller feature set decreased performance for binary classification but increased the discriminative power in multi-class classification and a very interesting insight is the discriminative information in PDM and PDD features. To our knowledge, the correlation between phase features and stød has not been studied

before. If PDD models noise in the voice source, it should be a salient feature for stød detection and this has been verified by the feature ranking experiment.

Phase features seem to be more informative for discrimination between all phones than some $\Delta$ and $\Delta\Delta$-derivatives of PLP features. Derivative features of MFCC or PLP coefficients are theoretically designed to model the speed and acceleration of speech organs such as tongue, lips, teeth etc., which means they are designed to model the filter in the source-filter model. Creak is produced in the vocal source and the features should therefore not be very informative for stød detection as the evolution of the filter does not impact the shape and movement of the source.

Select+ includes the first 4 MFCC and PLP coefficients because they correlate with stød to some extent as shown in Section 4.2.4. A similar discovery was made in Yoon et al. (2006) where creak could be predicted with standard PLP features and this effect has been replicated in the phone discrimination experiments in Section 4.3.3. In like fashion, it should be possible to immediately implement stød in ASR by annotating stød in the ASR dictionary.

To a large extent, select+ can replace standard PLP features in the phone discrimination experiment. It could indicate that phase information can inform phone classifiers in ASR systems, especially for Danish where stød detection can be important.

PDD is an estimate of the noise of the vocal source. Harmonics-to-noise ratio has been used in previous studies of stød as an estimate of the same, but did not turn out to be salient in feature salience ranking. The implementation of harmonics-to-noise ratio in Praat might be the cause of this difference. The arbitrary choice of assigning -200 as harmonics-to-noise ratio value could be the reason why the feature is not salient. Simple re-estimation a lower bound for harmonics-to-noise ratio did not make harmonics-to-noise ratio salient, but should not be interpreted as evidence that the measure is irrelevant for stød, because the implementation of the harmonics-to-noise ratio extraction rather may be the cause rather than the measure itself.

With the exception of MFCC and PLP features, all features in select+ are sensitive to energy in higher frequency bands and stød seems to correlate with energies in higher frequency bands. We cannot conclude that energies at particular frequency bands indicate stød, but it is a commonality for the voice quality feature in select+ which suggest that there is a relevant signal.

### 4.4.2 Stød detection

Viewing sample weight in connection with the varied annotation, the sample weight increases when optimised on extended annotation. To verify the results, sample weight optimisation was repeated several times and while sample weights varied by 0.1, but the difference, i.e. a larger sample weight when using extended vs.

raw annotation held constant in all trials. Sample weight is a measure of how much information a single sample should contribute to a model. Sample weights that are smaller than 1 indicate that a class annotation is impure, i.e. training samples that are stød-less are labelled as stød-bearing. The larger sample weight for extended annotation indicates that more samples included by the extension are stød-bearing, than stød-less.

Several potential sources of class impurity exist. The heuristic alignment is not perfect and relies on the accuracy of the existing time-coded annotation. The impurity of the positive class can also be due to annotation convention. Stødbasis and segmental phonetics requires stød to be annotated on a phone. If an annotator hears stød that is smeared across phone boundaries, that annotator must make a choice between the two phones. That stød manifestation happens on the second phase of a syllable and that the second phase coincides with the last half of a long vowel or a sonorant following a short vowel was observed in Fischer-Jørgensen (1989). The raison d'être is that a sufficient amount of voiced material needs to be present for stød to manifest. Hansen (2015) argues that this condition should be viewed as a phonological or phonotactic constraint rather than phonetic because stød is not elided in speech with high rate-of-speech or whispered speech which precludes voiced segments. Hansen also cites Grønnum & Basbøll (2001):

> "... stød in a long vowel may extend into a succeeding sonorant consonant. Stød in a sonorant consonant may already begin during the last part of the preceding short vowel and may continue into a succeeding voiced sound as well. Nor does the stød phase, the creaky voice, have a very definite and fixed duration."

The above observations by Hansen and Grønnum, the training and development set evaluation and sample weight difference suggest that stød is realised earlier than the second phase of the preceding short vowel. More samples in the preceding short vowel are stød-bearing than stød-less and therefore sample weight increases. The graphs in Figure 4.10 do not corroborate this. The Area Under the Curve and F1 does increase, but the curves are almost completely overlapping. A potential source of error that all models share is the phone alignment.

If this is the source of error, more refined segmentation of stød manifestation is necessary than the heuristic alignment in Section 4.1.3 and the annotation extension in Section 4.3.1. Rule-based approaches such as including the second half of a vowel preceding a stød-bearing consonant sonorant or including parts of subsequent sonorants could be one approach. The training data would still be impure, but with a sufficiently large amount of training data, robust statistical models could potentially be estimated.

Alternatively, an alignment could be statistically induced, but this would require more data. To the author's knowledge, the available manually-annotated data has been included in this study. When faced with data sparsity problems, bootstrapping can be applied to iteratively generate additional annotated data

for model estimation. Choosing a small amount of seed data where stød is present, and subsequently adding samples that have been classified with a high degree of accuracy until convergence, is a much used approach in absence of large volumes of training data. The attraction of such models would be to investigate further the timing of stød manifestation in relation to underlying segments. The JHP data set is well equipped for this task. High confidence stød samples can be extracted using simple majority voting. The challenge will be 1) if there is enough data in JHP to seed a bootstrapping model, 2) if it will be possible to recognise stød where the stød-bearing segment is not in the seed data. The problem in a bootstrapping approach remains how to account for the impact of the underlying segment.

### 4.4.3 Stød detection by phone discrimination

The interesting application of stød detection is further insight into the nature of stød and the discrimination of stød-bearing and stød-less phone, which is also the interesting application in ASR. Training statistical models that jointly model phone and stød are very good at discriminating stød-bearing and stød-less samples if the training data is from the same domain as the test data.

Cross validation shows high accuracy scores for both annotation variants. Applying the trained models to JHP data results in slightly above-average evaluation taking variance into consideration. The results are interesting because implementation into a speech recogniser in this form is fairly straightforward. A proof-of-concept can be done by simply adding stød annotation to the pronunciation model and should this experiment show positive results, training acoustic models with new features can be compared to a baseline in a similar fashion to the discrimination experiment.

There are some important distinctions between the discrimination and detection experiments. Because of the One-vs-One evaluation scheme, it was possible to estimate SVMs with a radial basis function kernel. This kernel does not scale well to larger data sets and it was not possible to use a SVM with a radial basis function kernel in binary classification with the hardware available.

In the stød detection experiments, mean subtraction based on speaker, corpus and gender was performed, but none of the variants performed better than using a mean estimated globally across corpora. In future studies, a phone or utterance-based mean subtraction experiment should be conducted. That stød can be used to discriminate segments indicate, that useful information is contributed by stød, but the contribution by the underlying phone in binary classification confounded the statistical models.

### 4.4.4 Computational modelling of stød

The results show that it is necessary to treat stød jointly with the underlying segment and we believe the segmental distribution helps reduce the false discovery rate. The same cocktail of acoustic features

that signal stød can occur where stød is not perceived and there is no lexical function to fulfill. The true distribution of stød is not reflected in the binary classification experiments, but is present in the phonetic symbols and the de facto factoristion of stød classes into stød-bearing phones in the multi-class classification experiments are therefore beneficial.

We do not conclude that it is not possible to detect stød in audio and one avenue of research we can identify is to normalise the acoustic features based on a mean and variance estimated for each phone, e.g. estimate mean and variance on samples labelled [a] and use that to standardise features extracted for [a$^{?}$] or similar standardisation. This research is beyond the scope this thesis because we will not be able to apply the standardisation in ASR without predicting the phone first.

The feature overlap between select+ and standard ASR features, the poor binary classification results and the relatively good multi-class classification results suggest that the best way to integrate stød in ASR is to extend the acoustic feature vector input rather than adding a specific feature for stød and jointly model phone and stød.

## 4.5 Chapter conclusions

We have ranked a large number of acoustic features for salience to stød detection in both elicited (Parole48 and DanPASS-mono) and non-elicited (JHP) speech and have found a set of 17 features – select+ – that signal stød manifestation particularly well. One of the interesting insights is the indication of phase features as important for stød detection and phone classification which we belive to be novel.

In the binary classification experiments, we can obtain the same classification accuracy using select+ as we can with all 120 features. However, reliable stød detection was not possible when formulated as a binary classification task using the available training and test data. The classifiers were not able to estimate a good model because the features do not sufficiently separate stød-bearing and stød-less classes in the feature space. If we define stød detection as discriminating between stød-bearing and stød-less variants of the same phone, i.e. a multi-class classification task, the detection of stød becomes possible and makes using stød in downstream NLP systems feasible.

The discrimination experiments indicate that PLP features carry a signal that indicates he presence of stød. This discovery faciliates prototyping speech recognition systems that model stød, because we can add stød annotation to the phonetic dictionary and compare performance to a baseline. If we wish to add voice quality features from select+ to the acoustic feature input, we will need to design a more advanced experiment that aligns the features.

# Chapter 5

# Modelling stød in automatic speech recognition

The intended application for the stød detection experiments in Chapter 4 is automatic speech recognition (ASR). Stød has a distinguishing lexical function and to implement and exploit this function in ASR is the objective of the experiments reported in this chapter. In the previous chapters, we have confirmed our assumptions on stød, namely that stød annotation is reliable, that we can use stød annotation to discover features that signal the presence of stød and that we can detect stød from acoustic features. The last assumption was only partially confirmed because it was necessary to predict phone and stød jointly.

ASR systems can model stød in the acoustic model (AM) only if the phone set includes stød-bearing phones. The studies in Chapter 4 demonstrated that a support vector machine with a radial basis function kernel trained on Perceptual Linear Perception (PLP) features can discriminate between stød-bearing and stød-less phone variants. Using the select+ feature set improves classification accuracy on semi-fine IPA annotation and using a coarser-grained set of classes, select+ performs well, but is outperformed by both PLP features and the full feature set. The conclusion is that stød detection is possible using standard ASR features, but can potentially be improved with voice quality features.

This chapter presents the development of a baseline ASR system as well as experiments where stød is integrated into an ASR system. The purpose is to implement and exploit stød using conventional ASR tools and techniques and the experiments entail adding stød annotation to the phonetic dictionary and extending the feature input with pitch-related features. Extending the phone set should be sufficient because the classifiers in Section 4.3.4 were able to discriminate stød-bearing and stød-less phones using standard ASR features.

There is little existing publicly-available research on or resources for Danish ASR. In a white paper on the state of Danish language technology and NLP (Pedersen et al., 2012), the quality of speech technology was not ranked due to disagreements between researchers and industry, and the availability of speech technology is ranked as poor or fragmented. Danish speech corpora are ranked as medium quality, with low coverage and maturity. The existing corpora we know of that can be used to train ASR systems are subject to access barriers. DanPASS, DK-Parole and LANCHART are not publicly available, and EUROM1 and Aurora2-3 can be acquired for a fee. The white paper concludes that support for speech technology as a whole is fragmentary.

Fortunately, ASR toolkits can be shared across natural languages and there are open and freely available ASR toolkits such as Sphinx (Placeway et al., 1997), Kaldi (Povey et al., 2011), the Hidden Markov model toolkit (Young, 1993) and RASR (Rybach et al., 2011) to name a few. The toolkits are based on machine learning techniques and can therefore be trained as long as data is available and contain scoring software to evaluate performance.

Though DK-Parole is a single speaker corpus and Aurora3 only contains spoken digits, ASR systems have been trained on these corpora (Henrichsen & Kirkedal, 2011; Kirkedal, 2013; Rajnoha & Pollák, 2011). These systems are academic systems for restricted domains (speaker-dependent ASR and spoken digits in noise) and the Word Error Rate (WER) performance is summarised in Table 5.1.

| Corpus | Task | %WER |
|---|---|---|
| DK-Parole (Henrichsen & Kirkedal, 2011) | Single speaker | 5.7 |
| Aurora3 (Rajnoha & Pollák, 2011) | Spoken digits | 24.39 |

Table 5.1: Published ASR evaluations for spoken Danish.

The methodology or *recipe* for training these systems is unavailable and the results might not be reproducible, which is necessary for meaningful comparison to the present work. To facilitate reproducible research, we have added the recipe developed for the experiments in this chapter in Appendix B.2.1. To develop the recipe, we first developed a Danish ASR system that does not model stød, which we use as a baseline to evaluate the influence of adding stød.

We wish to experiment with both standard GMM-based ASR systems and systems that make use of AMs based on neural networks. Of the ASR toolkits mentioned above, Kaldi is distributed as open source under a permissive license, has the necessary code to train deep neural network (DNN) AMs and contain several recipes describing methodologies for training ASR systems on English, Arabic, Czech etc. for a variety of tasks. We use recipes for similar corpora as inspiration for the baseline system.

To train ASR systems and especially DNN AMs, a large amount of data is required – more than is available in DK-Parole and DanPASS (LANCHART is sufficiently large). It turns out that a large Danish corpus that was not listed in Pedersen et al. (2012) exists. The Norwegian National Library Service hosts a large public domain corpus of read-aloud speech that also contains a Danish part. The corpus – Språkbanken – is large enough that it is possible to train DNN AMs and because the speech genre is read-aloud speech, it is easier to work with than LANCHART.

We describe the Språkbanken corpus in Section 5.1 and the development of an open source Danish ASR system using Kaldi and Språkbanken in Section 5.2. The recipe described in this setup forms the basis for all subsequent experiments. Baseline evaluation and experiments with stød modelling will be reported in Section 5.3 and the results analysed in Section 5.3.5. Section 5.4 will discuss the insights from Section 5.3.5 on acoustic and language modelling as it relates to stød.

## 5.1 The Språkbanken corpus

Nasjonalbiblioteket (The Norwegian National Library service) hosts a multilingual, multi-modal speech corpus known as Språkbanken[1]. The Danish part of the database contains read-aloud speech covering 7 regional dialects in Denmark as well as ages ranging from 18-70. The database was made available without restrictions in 2011.

The data was acquired by Nasjonalbiblioteket through the liquidation of Nordisk Språkteknologi (NST) in 2003. The data has been validated by NST in collaboration with Centre for Language Technology at Copenhagen University, but due to the liquidation and subsequent acquisition process, very little corpus description is available and most of the information is contained in a short description and analysis of the database. A considerable amount of manual work has been devoted to discover the actual structure of the corpus and convert data to a format consumable by ASR toolkits. This was also the case for the Swedish part of the Språkbanken corpus in Vanhainen & Salvi (2014). In mid-2015, the corpus was republished[2] by Nasjonalbiblioteket along with other linguistic resources under the name Språkbankens ressurskatalog (Språkbanken's resource catalogue) to make the resources easier accessible from the web, but the publication did not include a thorough description or restructuring of the data. The corpus description is based on Andersen (2008) and our own analysis of the corpus.

[1] Not to be confused with the text corpora and language resource centre at University of Gothenburg with the same name.
[2] Published using the CC0 description: `https://creativecommons.org/choose/zero/`

### 5.1.1 Speech data

All parts of the Språkbanken corpus consist of a database of 16 kHz recordings for ASR purposes, a 22 kHz database for dictation[3]. For the baseline system, only the 16 kHz database was used and summary corpus statistics can be seen in Table 5.2.

| Purpose | Speakers | Recordings | Hours |
|---|---|---|---|
| Training | 560 | 174720 | 316 |
| Test | 56 | 55272 | 77 |

Table 5.2: Summary table for the Språkbanken corpus.

The recordings were made in a closed office environment and are purportedly based on a *phonetically-balanced* manuscript, though this concept is not further defined in the description. Each line in a manuscript corresponds to a recording and each line can be a sentence, sequence of numbers, names, acronyms, single words or letters (i.e. the spelling of a word). According to the description, there is a systematic naming and structuring of the corpus. It was our experience that the description is not consistent with the actual structure and some directories that should contain data were in fact empty and the naming convention changes. The variation includes capitalisation and missing subdirectories.

Information about utterance, speaker, recording and manuscript identity was recoverable from a combination of file names, directory names and transcription files. The transcription files contain timed transcriptions for all lines uttered by a single speaker. These files also contain microphone specification, recording date and speaker information such as speaker id, full name, age, gender, region of youth, region of dialect, but across recordings, age and gender information is not consistently present in the transcription files.

The sound files are reported to be raw, linear PCM encoded 16 bit/16 kHz stereo recordings and prepended and appended with 100-200 ms silence. The files are actually NIST SPHERE files and 64 files are corrupt. According to Andersen (2008), the quality control has been thorough and consistent, though currently undocumented. Interestingly, Andersen notes that creaky voice can be audibly heard in the Danish speech data, but is not explicitly annotated.

The 7 regions used in the corpus are listed in Table 5.3. The regional division diverges from common regional dialect distinctions at the time of corpus creation, e.g. Eastern Danish as spoken in Scania and Bornholm is not accounted for and this dialect is considerably different from Copenhagen and Zealand due to influence from Swedish. Andersen comments that the division may have been commercially motivated.

[3]The 8kHz database disappeared in the 2015 republication.

| Region | English translation |
|---|---|
| Københavns-området | Copenhagen area |
| Sjælland utenom København | Zealand outside of Copenhagen |
| Fyn | Funen |
| Nord-Jylland | Northern Jutland |
| Vest-Jylland | Western Jutland |
| Øst-Jylland | East Jutland |
| Sør-Jylland | Southern Jutland |

Table 5.3: Dialect regions in the Danish part of Språkbanken.

### 5.1.2 Text data

Only orthographic transcriptions are contained in transcription files and no phonetic dictionary was available when the recipe was originally designed. In the 2015 launch of Språkbankens ressurskatalog, a phonetic dictionary and word n-gram frequency lists from NST were made available in the public domain. Unfortunately, it was not discovered in time to include the phonetic dictionary, although we where able to experiment with the n-gram lists in Section 5.2.4.

## 5.2 Recipe

In Kaldi, a recipe is a Bash file with a number of commands that can be executed to train an ASR system on a data set. The recipe described in this section is published in the Kaldi project under the recipe *sprakbanken*. The recipe downloads the Danish 16 kHz part of the Språkbanken corpus described in Section 5.1, a phonetic dictionary, a *phone set specification* and linguistic questions used to model pronunciation. These resources are described below.

### 5.2.1 Textual preprocessing

The training transcripts are stripped of punctuation except at the end of words that are followed by an uppercase letter. We preprocess numbers to spoken form e.g. `310` to `TRE HUNDREDE OG TI` and `12.` to `TOLVTE`. We try to discriminate between cardinal and ordinal numbers based on the final punctuation. At the core of the method is a lexicon that maps numbers 0-100, 1000, 2000-2020, ordinals 1-30 and splits numbers written as a single word, e.g. `TREHUNDREDEOGTI` to `TRE HUNDREDE OG TI`. The rule-based normalisation detects Danish social security numbers (CPR numbers), numbers in the thousands, hundreds

or tens and numbers with a leading zero (used in e.g. telephone and CPR numbers). The rules decompose numbers left-to-right and apply the lexicon first to thousands, then to hundreds and finally tens. The proper spoken connectives (either `OG` or $\emptyset$) are also added.

We convert all text to uppercase and expand abbreviations in the corpus using a dictionary plus rules encoded as regular expressions. The remaining punctuation is used to split sentences. Annotation included in the transcriptions are removed such as tabs or 4+ whitespace sequences between single letters (denotes spelling). Dates and special characters like % and + are also converted to spoken form. Finally, the text is converted to UTF-8 encoding.

#### 5.2.1.1 Phonetic transcription

The training transcripts contain approximately 66000 unique words and since we did not have access to the phonetic dictionary mentioned in Section 5.1.2, we instead used automatic transcription to generate the phonetic dictionary. Automatic transcription is flexible and therefore it will be easy to extend the phonetic dictionary with new entries and to experiment with stød annotation. Two systems were available: eSpeak (Duddington, 2012) and Phonix (Henrichsen, 2014). eSpeak is a shallow system that primarily uses an internal dictionary and letter-to-sound rules to generate a phonetic transcription using ASCII-IPA. Phonix was used to transcribe LANCHART data and uses a fallback strategy where the first strategy is a dictionary lookup, the second is compound splitting followed by another lookup in the same dictionary, and finally letter-to-sound rules, i.e. a deeper linguistic analysis than used by eSpeak. Phonix models phonological factors, e.g. that [p], [t] and [k] is pronounced as [b], [d] and [g] respectively if the consonants are not word initial which eSpeak only does if the word is in the internal dictionary.

In Kirkedal (2014), we compared the two automatic transcribers on DK-Parole where the speech genre is similar to Språkbanken. The conclusion of the study is that eSpeak produces better phonetic transcriptions for ASR dictionaries than Phonix. One reason why Phonix did not perform well is a fallback strategy that generates a phonetic representation where the word is spelled out. This strategy caters to a text-to-speech scenario but reduces performance in ASR because the fallback strategy is often triggered on long unknown words (or compound words where compound splitting fails) where the number of symbols in the phonetic transcription will be greatly different from the uttered phones. eSpeak does not have the same fallback strategy but uses letter-to-sound rules. Based on these results, eSpeak is used in the experiments in Section 5.3.4.

**Preprocessing phonetic transcriptions**

eSpeak transcribes words using phonetic symbols from ASCII-IPA and also performs language identification at word level and if words are identified as e.g. French, the word is transcribed using the French transcriber. In the recipe, the language ID and diacritic markers are stripped from the phonetic symbols before they are tokenised.

Because of the language ID feature in eSpeak, some phones occur very rarely because they represents sounds from a foreign language and do not occur in Danish. Very rare phones must be mapped to a similar phone symbol from Danish with sufficient representation in the data and e.g the dark [L] phone in English transcriptions are mapped to the normal [l] phone. After transcription, the words input to eSpeak are uppercased.

We create a phone set specification with all the phonetic symbols used in the dictionary where variants of a so-called *base* phone is mapped to each other, e.g. [e], [E] and [i] in Table 5.4 are base phones. This terminology is motivated by implementation details rather than theoretical reasons, because the phone set specification ensures that variants of a base phone share the root node of a phonetic decision tree (PDT). Phones with low occurrence in the corpus and stress-dependent versions of base phones are on the same line to cluster similar phones to each other. Table 5.4 shows 3 lines from the specification. The phone [I] occurs less than 10 times in the training data, which is too infrequent to estimate a separate phone model and we therefore map [I] to [i]. Similarly, stress-marked variants of base phones are mapped to their unstressed variant.

| Phone | Phone alias |
|---|---|
| e | e 'e |
| E | E 'E |
| i | i 'i I 'I |

Table 5.4: Example from the prepared phone specification.

To be able to distinguish the phones in a root node, a suitable question must be available in the PDT building process. The questions needed to distinguish the phones in Table 5.4 can be seen in Table 5.5. If distinguishing between ['i] and [i] sufficiently increases the log-likelihood of the data under the assumption that the data on each side of the split is modelled by a single Gaussian, the algorithm splits the data accordingly.

| Question | Phone set |
|---|---|
| Stressed? | `'E 'e 'i 'I` |
| Segment=I | `I 'I` |
| Segment=i | `i 'i` |

Table 5.5: Example questions that makes it possible for the PDT algorithm to distinguish stressed variants of the base phone during tree building.

The preprocessing of phonetic transcriptions is a time-consuming trial-and-error process that entails mapping infrequent phones in the phone set specification, adding linguistic questions to the PDT algorithm and transcribing and preprocessing the phonetic dictionary. We have compiled a dictionary, phone set specification with phone mapping and linguistic questions in the public recipe so future users do not need to devote time to develop these lexical resources.

The public recipe and the experiments in this chapter do not model stress in the phonetic dictionary. Stress-annotated variants are included in the phone specification and the necessary question to distinguish the phones is also added to facilitate extension with stressed phones as shown in the example in Table 5.4 and 5.5, but because stress is not in the phonetic dictionary, stressed variants will not be observed in the data and the PDT clustering algorithm is prevented from splitting a node into stressed and unstressed variants by a minimum-state occupancy threshold.

After preprocessing, the prerequisite resources are in a format consumable by Kaldi programs and can be compiled into a finite state transducer (FST) representation.

### 5.2.2 Data sets

The Språkbanken subcorpus Testing in Table 5.2 contains 3 subcorpora: Stasjon03, Stasjon05 and Stasjon06, respectively containing approximately 16.5 hours, 51 hours and 9.5 hours. Compared to existing test sets for Mandarin (Hwang et al., 2004) or English such as 2001 Hub5, Wall Street Journal (WSJ) test sets (Paul & Baker, 1992) and the newer test sets created for the Librispeech corpus (Panayotov et al., 2015), the Språkbanken test sets are 15 times larger. While reserving approximately 20% of the available data for testing purposes is standard in machine learning methodology, ASR test and development sets are frequently 5 hours long or shorter and the training data is much larger than the training section in Språkbanken. We therefore add Stasjon05 to the training data, while Stasjon03 and Stasjon06 are reserved for development and testing purposes, respectively. The new partitioning of Språkbanken can be seen in Table 5.6 including the purpose of the data set, duration, type and token counts.

| Data set | Purpose | Hours | Types | Tokens |
|---|---|---|---|---|
| train | training | 367 | 65667 | 2366183 |
| train_120kshort† | flat start training | 93 | 27103 | 591423 |
| Stasjon03 | development | 16.5 | 7311 | 112062 |
| Stasjon06 | testing | 9.5 | 5683 | 72987 |

Table 5.6: Word type and token counts for the data sets in the sprakbanken recipe. Note that the *train* set is different from Train in Table 5.2. The symbol † denotes that *train_120kshort* is a subset of *train*.

The Stasjon03 and Stasjon06 data sets are still comparatively large and Stasjon03 is currently not a part of the public recipe but was used in Sections 5.2.1.1, 5.2.4 and 5.3.3, and may in the future find other uses such as discriminative training with Maximum Mutual Information or Minimum Phone Error rate.

When estimating an AM from a flat start, it is standard methodology to use the shortest utterances from the training set[4]. While this reduces the amount of training data used to estimate the monophone model, it is easier to induce an accurate alignment from a flat start if the utterances are short and the entire training set can be used at a later stage to train triphone ASR systems. A training subset – train_120kshort – containing the 120000 shortest recordings from train is created to induce a flat start alignment and a monophone system.

### 5.2.3 Feature sets

Feature extraction follows the same procedure whether we extract MFCC or PLP features. We extract feature vectors as described in Section 2.5.1 and discard feature vectors if the 25 ms context window exceeds the end or beginning of a recording. Subsequently, we apply speaker-based cepstral mean and variance normalisation to feature vectors. Cepstral mean and variance normalisation can degrade performance when estimated per-utterance on short utterances because there is no distinction between noise, silence and speech when the mean is estimated (Togneri et al., 2006) and there may not be sufficient data in a short utterance to estimate a good long-term mean (Prasad & Umesh, 2013). We normalise per-speaker to avoid problems of data sparsity and because if normalisation is applied per-utterance, speaker information will be lost and the transformation used for speaker adaptation will be estimated on different off-sets.

[4]See e.g. the WSJ and Librispeech recipes.

### 5.2.4 Language models

Because the genre is read-aloud speech, many occurrences of the same sentences are in the transcriptions. For language modelling purposes, a set with unique sentences are compiled that are disjoint from Stasjon03 and Stasjon06, which we use for evaluation. We use the IRSTLM toolkit (Federico et al., 2008) to train a trigram Arpa LM on the unique sentence transcripts.

We also estimated a trigram LM on the n-gram frequency lists mentioned in Section 5.1.2. We used the SRILM toolkit (Stolcke, 2002) to generate the Arpa format LM on the frequency lists and normalised n-grams using the approach described Section 5.2.1. We used SRILM because the toolkit has a built-in function to estimate an LM from a mix of text and frequency lists and can automatically merge n-grams that become identical after normalisation.

Arpa LMs are converted to weighted FST acceptors using the `arpa2fst` program distributed with Kaldi. To determine which LM to use in our experiments, we evaluate the WER performance of four ASR systems trained using LDA-projected MFCC features and speaker adaptive training: two GMM-based systems and two DNN-based systems, that use either the frequency list LM estimated with SRILM or the transcript LM estimated with IRSTLM. We use Witten-Bell smoothing (Witten & Bell, 1991) to smooth frequency counts. Witten-Bell is a standard technique in ASR (Stolcke et al., 2000; Matějka et al., 2006), the default in Kaldi and recommended for short texts because the smoothing works well in many conditions and is less sensitive to the amount of data used to estimate the LM and the text type (Federico & De Mori, 1998).

The results in Table 5.7 indicate that while the unique transcript LM is estimated on a small amount of data (approximately 1.3 million sentences), it is *narrow* in the sense that it very successfully models the text domain. The n-gram frequency lists are estimated on a much larger newswire text corpus (approximately 290 million sentences), but does not match the data.

| LM | AM type | %WER |
|---|---|---|
| Frequency list | GMM | 41.55 |
| Frequency list | DNN | 30.01 |
| Unique transcript | GMM | 20.71 |
| Unique transcript | DNN | 15.78 |

Table 5.7: ASR performance on Stasjon03 data for systems using an LM estimated on NST n-gram frequency lists and on unique transcripts. The results are not directly comparable with later results because a newer version of Kaldi was used for this set of experiments.

This is counter to our expectation and we manually inspected the output. All data sets in Språkbanken contain utterances that can be categorised as follows:

1. Named entity utterance: Town names, street names, utterances with first name last name e.g. ODENSE, TOMMY ANDERSEN, DELOITTE
2. Repetition utterance: The same word repeated three time, e.g. PAPIR PAPIR PAPIR
3. Spelling utterance: Spelling a word aloud
4. Number utterance: Dates or a string of number such as 3546 8917 200
5. Sentence utterance: Utterance that contain grammatical utterances and spoken punctuation

We observed more errors in the decoding output on utterances in categories 1-4 when we use a LM estimated on the larger newswire corpus and we conjecture that newswire text does not match the domain because these types of utterances are overrepresented in the Språkbanken corpus as opposed to newswire text.

Because we study stød in connection with acoustic and pronunciation modelling, we want a LM that fits the domain well to ensure that we evaluate against a strong baseline. We will therefore use the transcript LM in the following experiments. We will refer to the LM as *3g*.

### 5.2.5 Training acoustic models

A bootstrapping process where several systems are estimated one after another is used. The WSJ and Librispeech recipes in Kaldi have served as templates for this recipe and the upper bounds on the number of Gaussians and leaves used in the recipe are reused here with the exception of the GMM-based tri4b and DNN-based nnet5c systems, where we tuned the training parameters. We chose this approach because the same parameters give state-of-the-art results on WSJ and are used for Librispeech too though it is many times larger than WSJ.

#### 5.2.5.1 GMM AMs and feature transforms

Initially, a flat start context-independent monophone AM – mono0a – is trained for 40 iterations on train_120k using Viterbi aligment. We use 40 iterations as it is the standard in 24 out of 29 recipes including TIMIT, Switchboard, WSJ, TEDLIUM, Librispeech and GlobalPhone, i.e. across different corpus sizes and languages and the primary function of a monophone model is to generate an alignment that can be used to train context-dependent AMs. 39-dimensional feature vectors with first and second order derivative

features, 5-state HMM topology for silence phones and 3-state topology for 'non'-silence phones (standard in Kaldi) are used to model phones with at most 1000 Gaussians. Kaldi treats context-independent systems as a special case of context-dependent systems where the phonetic context is 0 phones to the left and right and modelled with a trivial decision tree with no splits. Because no states are tied, GMMs are mixed up from the number of context-independent phones until we reach a total of 1000 Gaussians.

The *Gaussians* parameter is a total across all GMMs and is increased by *mixing up* Gaussians during training. The number of Gaussians in a GMM is based on a data count $\gamma$ that is raised to the power of 0.2 and if the alignment maps e.g. 5000 frames to a GMM, that particular GMM will have 5 Gaussian components ($\gamma^{0.2} = 5000^{0.2} = 5.49$ rounded) when training finishes. A schedule is computed so the total number of Gaussians increase by a fraction each iteration, e.g. if we increase the number of Gaussians for 20 iterations from 2000 to 10000, then $\frac{10000-2000}{20} = 400$ extra Gaussians will be estimated each iteration by splitting the Gaussians estimated in the previous iteration that have the highest occurrence count.

We follow standard methodology and use only the shortest training samples to train the monophone alignment. There is little literature on the subject of audio length in flat start alignment estimation but both the CMU Sphinx[5] and the Kaldi recipe for the Wall Street Journal corpus[6] use this methodology[7].

The monophone AM is used to align and segment the entire training set. Based on this alignment, a context-dependent triphone AM – tri1 – is trained on the full training data. The PDT is restricted to 2000 states and 10000 Gaussians. The *states* parameter ocrrespond to the number of leaves in the PDT or rather the number of GMMs estimated.

Based on an alignment created by tri1, tri2a is trained using 2500 states and 15000 Gaussians.

tri2b is also based on an alignment created by tri1, but a LDA feature space transform is applied to the feature vectors. Each feature vector is extended with the feature vectors in the left and right context. For tri2b, a window of 11 frames are used resulting in $11 * 13 = 143$ dimensions which are projected into 40 dimensions using Maximum Linear Likelihood Transform (MLLT). Training differs slightly from previous steps. First, a transformation matrix $\mathbf{M}$ that maps from the original 143-dimensional feature space to the 40-dimensional LDA space is estimated, then – at given intervals – we estimate a new MLLT matrix $\mathbf{T}$ on a subset of training data and use $\mathbf{T}$ to update the model means and $\mathbf{M}$. The system is trained for 35 iterations with the same restrictions on number of states and Gaussians as tri2a, and $\mathbf{M}$ is updated four times during training.

[5] `http://cmusphinx.sourceforge.net/wiki/tutorialam` (Data preparation section)

[6] `https://github.com/kaldi-asr/kaldi/blob/master/egs/wsj/s5/run.sh`

[7] Discussions on the Kaldi help forum also states that this is helpful: `http://kaldi-asr.org/forums.html?place=msg%2Fkaldi-help%2FR0ao4x5qDZ0%2FFzybtcMYCAAJ`(the link loads slowly).

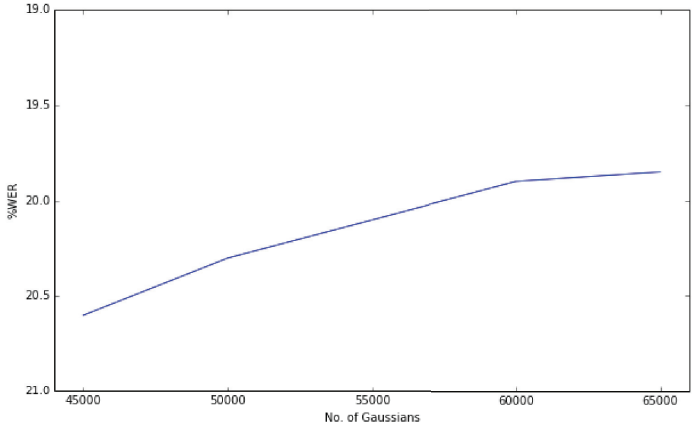



Figure 5.1: WER performance as a function of the total number of Gaussians in GMMs trained on LDA speaker-adapted features derived from MFCC.

tri3b is based on a tri2b alignment and estimates an AM using speaker-adaptive training (SAT) which normalises intra-speaker variability. Speaker-specific means and variances are computed using Constrained Maximum Likelihood Linear Regression (cMLLR, also known as fMLLR) on top of LDA features computed for tri2b. Four cMLLR updates are applied instead of MLLT, but otherwise settings are identical.

Two systems are trained on tri3b using speaker-adaptive training and with a larger threshold on PDT clustering and state-tying. For tri4a, the number of states are increased to 4200 and number of Gaussians to 40000. For tri4b, max number of states is 4800 and Gaussians 60000. In this toolkit, the PDT is grown until the specified maximum number of leaves are obtained, and subsequently the states are clustered and tied[8]. State-tying reduces the number of pdfs to 900 and 1100 for tri4a and tri4b respectively, which is twice the reduction compared to tri2b and tri3b (450-500). 23% leaf states can be tied without reducing the likelihood of the data significantly and we therefore do not increase the upper bound further.

To determine the number of Gaussians, we increased the total number of Gaussians from 40000 (tri4a) to 65000 in steps of 5000 and all systems except tri4a use 4800 states. In Figure 5.1, we see a stagnation

[8]This is unlike Sphinx-3, where the tree is fully grown and then reduced to the specified number of leaves.

in WER improvement on Stasjon03 above 60000, and stopped to prevent overfitting. The parameters and feature types for the mentioned systems can be seen in Figure 5.8.

| System | Feature type | #Leaf states | #Gaussians |
|---|---|---|---|
| mono0a | $\Delta$ and $\Delta\Delta$ | #Phones | 1000 |
| tri1 | $\Delta$ and $\Delta\Delta$ | 2000 | 10000 |
| tri2a | $\Delta$ and $\Delta\Delta$ | 2500 | 15000 |
| tri2b | LDA-transformed | 2500 | 15000 |
| tri3b | LDA and SAT-transformed | 2500 | 15000 |
| tri4a | LDA and SAT-transformed | 4200 | 40000 |
| tri3b | LDA and SAT-transformed | 4800 | 60000 |

Table 5.8: AM parameters and feature types for GMM-based systems.

#### 5.2.5.2 Neural network AMs

We use the built-in `train_tanh_fast.sh` script to train an DNN AM. The DNN has 5 hidden layers with tanh nonlinearities and is computed based on an alignment by tri4b, i.e. the AM is trained on the same 40-dimensional features that have been transformed with LDA, MLLT and SAT. The hyperbolic tangent (tanh) function is an S-formed function similar to the sigmoid function, that outputs values in the range [-1,1] (sigmoid function output values in [0,1]). Because we need speaker-adapted and LDA transformed features, it is necessary to reuse the cMLLR transform generated during decoding with tri4b when training and decoding using the DNN. An illustration of the DNN can be seen in Figure 5.2.

The input layer *Input* consists of a *SpliceComponent*, where input features are spliced in a context window of $\pm 4$, i.e. the input vectors are 40 dimensional and the output vectors are 360-dimensional. The 360-dimensional vectors are decorrelated or "whitened" using MLLT (Rath et al., 2013) in a second component and expanded to 1024 dimensions. The last component in the first layer is known as a *PreconditionOnline* component and has the same input and output dimensions, in this case 1024. The component is used to estimate an input and an output matrix used by the special Natural Gradient - Stochastic Gradient Descent algorithm (Povey et al., 2014). Rather than a scalar learning rate, the algorithm uses a matrix of learning rates, one for each dimension that is estimated for each mini-batch. The motivation for different learning rates in different dimensions is to control instability and prevent parameters from exploding or vanishing during training. All hidden and visible layers have a preconditioning layer as the last component.

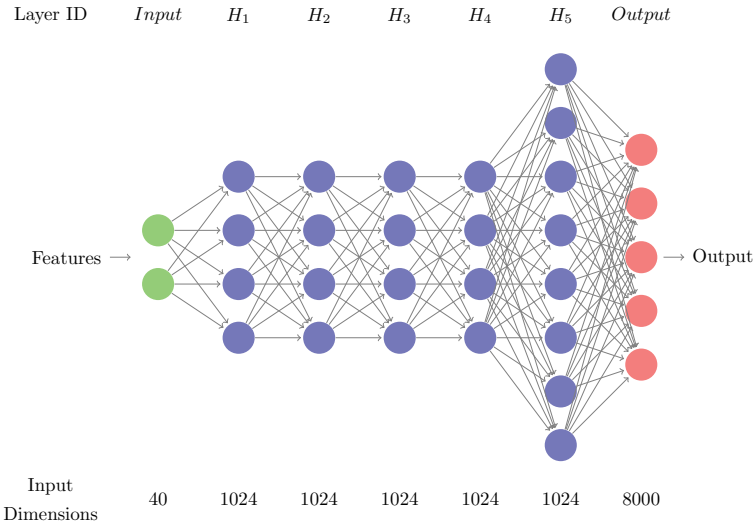



Figure 5.2: Architecture of the neural network AM. The number of neurons in a layer only serves to show relative layer size, i.e. the input layer is smaller than the hidden layers and $H_5$ is largest. Note that the *input* dimensions are specified in the bottom of the figure and they correspond to the layer size of the *previous* layer The output dimension at the output layer depends on the phonetic dictionary and the phonetic decision tree.

The five hidden layers $H_1 \ldots H_5$ consist of a tanh component and a preconditioning component and all take as input 1024 dimensional vectors and all except the last layer output 1024 dimensional vectors. $H_5$ outputs 8000-dimensional vectors and is known as a *mix-up* layer analogous to mixing up in GMM-based AMs. In the GMM-based setup, the data points used to estimate a Gaussian are split in two. The means of the two Gaussians are perturbed in either direction and training continues. In DNN training, rows of the weight matrix is split and perturbed to create virtual output targets that can be mapped to each leaf in the decision tree.

The 8000-dimensional vectors are passed to the final layer to a *Softmax* component that uses an activation function that converts input vectors to a value between 0 and 1 and a *SumGroup* component sums the correct virtual targets to the targets that correspond to leaf nodes in the decision tree.

The number of layers and layer size is chosen based on the guidelines in Kaldi which specify layers

based on training data size, e.g. 2 layers for 3 h and 4 layers for 100 h. Each hidden layer in nnet5c has 1024 nodes with tanh non-linearities. The DNN is trained for 20 epochs equalling 700 iterations and has a total of 12768064 parameters. 30 CPUs running parallel training jobs on approximately 200000 samples per iteration can estimate the DNN AM in 7-8 days.

We experimented with 4, 5 and 6 hidden layers and were forced to stop at 5 layers because training was too slow on the available hardware. Similarly, we experimented with hidden layer sizes of 512, 1024 and 2048 and chose 1024 because we faced similar problems with larger layer sizes. Adding the fifth hidden layer improved performance by 2-2.4 points absolute. The effect of mixing-up in the output of the final hidden layer was in the magnitude of 0-0.1 WER absolute, but we chose to include it because not mixing-up could have negative effects whereas mixing-up did not.

The training script uses an initial learning rate of 0.001 and reduce it to a final learning rate of 0.01. We set this rate based on recommendation in the Kaldi scripts.

### 5.2.6 Comments on the Kaldi toolkit

An early version of the recipe presented above is included in the Kaldi repository on GitHub[9] and the recipe used for the experiments in this chapter is included in Appendix B.2.1. To repeat the experiments in this chapter, replace the `run.sh` file in the sprakbanken setup in Kaldi with the commands in Appendix B.2.1.

When building the phonetic decision tree, it is possible to force phones to share a *probability distribution function* (pdf) and this is by default the case for phones that model silence, but can also be applied to e.g. stressed or tone-dependent phones (and stød-bearing phones). Because it is not always possible to uniquely map between a phone and a pdf, the AM outputs a *transition-id* to the decoding graph instead of a pdf. A transition-id can be mapped to a pair:

1. transition-state
2. transition-index

The *transition-index* identifies a transition out of a transition-state and can also encode the destination state. The *transition-state* is mapped to a triple:

1. phone
2. HMM-state (0, 1 or 2 in a three-state HMM)
3. pdf

[9]The repository is at `https://github.com/kaldi-asr/kaldi`.

The terminology can be quite confusing and the transition-state is the important concept to understand. During training, the labels in the input sequence are transition-ids (one per vector) and via transition-states they identify the correct pdf.

Therefore, the descriptive power of AMs is influenced by the number of transition states and transition-ids as well as the number of Gaussians. Where a larger number of Gaussians increase the descriptive power of a GMM AM[10], more transition-ids and transition-states make the model more complex and harder to train. An increase in transition-states mean a larger number of transitions to train and an increase in transition-ids mean a larger input alphabet to the decoding graph, which will be the effect of annotating stød-bearing phones in the phonetic dictionary.

## 5.3 Experiment

The conclusion in Chapter 4 is that stød can be detected, directly integrated into ASR via the phonetic dictionary and standard ASR features can model the difference between stød-bearing and stød-less phones. In this section, two experiments based on the recipe in Section 5.2 are reported. We train two baselines following the described recipe on MFCC and PLP features, respectively. We add stød to the phonetic transcriptions in the phonetic dictionary and we train corresponding systems in a MFCC+stød condition and a PLP+stød condition.

Two additional experimental conditions – MFCC+stød+pitch and PLP+stød+pitch – add the pitch-related probability-of-voicing, log-pitch and $\Delta$log-pitch features to the acoustic feature input to see whether pitch information improves evaluation as suggested by the feature ranking experiment in Chapter 4.

In the following sections, *baseline* systems use phonetic dictionaries that do not model stød, *stød-informed* systems use dictionaries that model stød and *stød+pitch* systems are stød-informed systems that are trained on PLP or MFCC features extended with pitch-related features.

### 5.3.1 Adding stød

The amount of training data needed to train an ASR system cannot feasibly be transcribed by phonetic experts and the standard methodology in ASR training is to use a phonetic dictionary to model pronunciation. The transcription produced by eSpeak is less accurate than the manual transcription in DanPASS-mono, Parole48 and the JHP sample, but due to the increased amount of data, forced Viterbi alignment should perform well. However, the experimental setup does break with the approach used so far because we do not make use of manually transcribed data.

[10] See Section 2.5.3.2 for an explanation of descriptive power.

We change the post-processing of the phonetic transcriptions generated by eSpeak so the stød annotation ([?]) is not removed to generate two dictionaries that only differ with respect to stød annotation, i.e. the phonetic dictionaries used in both experiments are of the same size and pronunciation variants are not included.

The phone sets differ as a consequence of the added stød annotation which can be seen in the the phone set specification. The difference is illustrated in Table 5.9.

| Without stød annotation | With stød annotation |
|---|---|
| `e 'e` | `e ?e 'e '?e` |
| `E 'E` | `E ?E ?'E '?E` |
| `i 'i I 'I` | `i ?i ?'i'i I 'I` |

Table 5.9: Phone list comparison. Each cell contains a base phone and all variants of the base phone that occur in their respective phonetic dictionaries.

Stød-bearing phones are defined as variants of a base phone in the phone specification and must be split when growing the PDT to have separate models estimated. The tree growing algorithm must split the stød-bearing variant from the base phone for explicit stød modelling to have an impact, i.e. modelling stød must give the largest likelihood increase at some point.

eSpeak is not consistent in the order of assigning diacritics to a phone and two variants of the same phone can exist such as [`?'E`] and [`'?E`]. Instead of correcting this discrepancy, both versions are added to the phone specification to maintain the possibility to easily extend the dictionary with new transcriptions generated by eSpeak, but no questions to distinguish the two symbols are created and one model is used to model both phones.

In Table 5.10, the questions added to the PDT building process are the same as in Table 5.5 with the exception of the first question that distinguishes stød-bearing sonorants. As illustrated in the much larger phone sets, adding stød annotation increases the number of word-position dependent phones (from 320 to 448). The theoretical number of context-dependent triphones becomes $448^3 = 89,915,392$ phones. However, many word-position dependent phones and context-dependent triphones will not occur in the data[11], which is word-medial silence.) and will not be estimated due to the minimum state occupancy constraint in the PDT or the determinisation operation applied between WFST compositions, which means the actual number of triphones will be dramatically smaller.

[11] For instance, there are no occurences of [`SIL_I`]

| Question | Phone set |
|---|---|
| Stød-bearing? | ?@- ?& ?&+ '?& '?&+ ?O '?O ?A '?A ?d ?e '?e ?E ?'E '?E ?i ?'i ?m ?o '?o ?O '?O ?s ?u '?u ?V '?V ?W ?W+ '?W '?W+ ?y '?y |
| Stressed? | '?& '?&+ '@ '@- '& '&+ 'O '?O '3 'a 'A '?A 'aI 'e '?e ?'E '?E ?'i 'i 'I 'o '?o 'O '?O 'u '?u 'U 'V '?V 'W '?W '?W+ 'W+ 'y '?y 'Y |
| I? | I 'I |
| i? | i 'i |

Table 5.10: Examples of manually added linguistic questions for phonetic clustering. The phone sets only contain observed phones.

Though the number of phones increas as a consequence of stød, the maximum number of leaves and the maximum number of Gaussians are held constant between systems in different conditions. This does not force the parameters in the experiments to be the same, but imposes an upper bound. An upper bound is desirable because the descriptive power of a GMM AM is correlated with the number of Gaussians and states it can use to model non-normal distributions.

| Dictionary | baseline | stød | difference |
|---|---|---|---|
| Entries | 65667 | 65667 | 0 |
| Unique transcriptions | 64610 | 64951 | 341 |
| 4x | 5 | 0 | -5 |
| 3x | 54 | 27 | -27 |
| 2x | 930 | 662 | -268 |

Table 5.11: Statistics of explicit stød modelling. *2x* denote the number of phonetic transcriptions that can be mapped to two words, *3x* denotes the number of phonetic transcriptions that can be mapped to three words, etc. The third column is computed as $stød - baseline = difference$ and show the impact on the phonetic transcriptions in the dictionary.

Because the number of phones increases, several ambiguous phonetic representations are disambiguated. A stød-less phonetic transcription such as [vEr] from eSpeak represents four words: hver (EN: every), værd (EN: worth), vær (EN: be-*imperative*) and vejr (EN: weather). Adding stød partially resolves the homophony such that [vEr] represents hver and vær while [v?Er] represents værd and vejr. Table 5.11

shows the number of entries in the dictionaries, the number of unique phonetic representations, the degree of polygraphy (i.e. a homophonic representation is enumerated in 4x if it is the phonetic transcription of four different words, 3x if it represents three words, etc.). To show the utility of the disambiguation, the proportion of tokens in DanPASS-mono, Parole48 and Stasjon06 affected by the phonetic disambiguation is listed in Table 5.12 as well as the absolute difference.

341 homophonic transcriptions are resolved by adding stød, the most polygraphic phonetic representations (4x) are resolved, 3x is halved and 2x decreases by more than 28% even though polygraphy cascades from 4x to 3x to 2x (as in the case of [vEr] and [v?Er], which removes a count in 4x but adds 2 counts in 2x).

| Corpora | baseline | stød | difference |
|---|---|---|---|
| DanPASS-mono tokens | 8.9% | 1.9% | -7% |
| Parole48 tokens | 36.2% | 9.5% | -26.7% |
| Stasjon06 tokens | 36.7% | 7.7% | -27% |

Table 5.12: Statistics of explicit stød modelling on three other corpora. The third column is computed as *stød-baseline=difference* and show the number of tokens in the corpora that have an ambiguous phonetic representation and the absolute reduction as a consequence of explicit stød modelling.

### 5.3.2 Evaluation

The systems are evaluated using the widely used Word Error Rate (WER). WER is calculated as

$$WER = \frac{Deletions + Insertions + Substitutions}{N} \cdot 100\% \tag{5.1}$$

$N$ is the number of words in the reference. If more words are present in the ASR hypothesis than in the reference, WER can exceed 100% due to a high number of deletions. WER is a hard metric with a 0/1 loss function, i.e. the word is either identical or not. There is no distinction between falsely recognising figs/fix and figs/mountain – both count equally as an error.

WER is edit distance and a lower rate indicates a better result. Other metrics that are generalisations of WER such as Sentence Error Rate, Character Error Rate or Phone Error Rate has been proposed. Phone error rate has been successfully applied as the optimisation metric in Minimum Phone Error training.

To calculate the statistical significance of performance increase or decrease, we use the Matched Pairs Sentence-Segment Word Error (MAPSSWE) implemented by the National Institute of Standards and Technology (NIST) in the software package sctk (Fiscus, 1998, 2007) – more specifically the program called

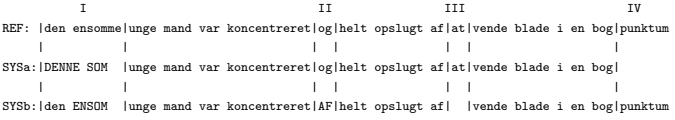



Figure 5.3: Example of MAPSSWE error calculation: The four segments I, II, III and IV are errorful segments and I counts as one error for both SYSa and SYSb, II and III count as errors in SYSb and IV counts as an error in SYSa.

`sc_stats`. The two-tailed test assumes that errors are normally distributed and a sufficiently large number of sentence segments are necessary for this assumption to hold (Gillick & Cox, 1989). We assume that Stasjon06 and Stasjon03 are large enough data sets that the normality assumption is reasonable because both data sets are larger than the NIST test sets the algorithm was designed for.

The illustration of the MAPSSWE method in Figure 5.3 show four errorful sentence segments. Segments can be of variable length and are specific to the pair of systems being evaluated. Segments must be bounded by at least two correctly recognised words in both systems, e.g. segment III is bounded on the left by opslugt af and by vende blade i en bog on the right. We can then expect the independence assumption to hold because the segment was recognised in the same linguistic and acoustic context. The number of words (two) is the segment boundary length because the LM used during decoding is a trigram model and therefore uses a two-word history to estimate the probability of the next word.

In most applications including medical dictation, translation dictation and respeaking, real-time ASR with incremental output is desirable, and the decoder must be able to decode as fast (or faster) than the speaker talks. A *real-time factor* (*RTF*) below 1 indicates that decoding is faster than real-time, while $RTF > 1$ is slower than real-time. A wide *beam* parameter increases the number of hypotheses generated during decoding and can slow down the recognition process, so RTF constraints effectively places an upper bound on beam sizes and other parameters that increase the complexity of decoding. A desirable RTF leaves a time buffer for potential network latency or similar and is therefore $RTF < 1$ rather than $RTF = 1$.

When comparing RTF, we use a single 32 core server with 64 GB that is exclusive to the test, i.e. the server is reserved, and to control for hard disk performance, memory consumption and CPU availability, only 7 parallel decoding processes (one per speaker) is running at the same time. We have also made sure that Stasjon06 and Stasjon03 fit in memory to avoid I/O bottlenecks.

#### 5.3.2.1 Equivalence classes

Stød-bearing phones can be mapped to stød-less phones, such as the base phone, during training. State-tying can cluster stød-bearing and stød-less phones if the decrease in the likelihood of the data is less than the threshold $S(o)$ (see Section 2.5.3.1 on page 30). If state-tying does not cluster training data for stød-bearing and stød-less phones, it supports the relevance of stød modelling in ASR because a) stød was chosen as a splitting criterion and b) state-tying did not merge the training data after growing the PDT.

The transition-states map between word-position dependent phones, HMM states and pdfs, i.e. which states are tied, and by proxy, which phones are modelled by the same pdfs. We define *phone identity* between two word-position-dependent phones as phones where the ordered set of pdfs is identical[12] An *equivalence class* is therefore a set of phones that share phone identity. An *independent* equivalence class is defined as a set of phones where all phones in the set are stød-bearing. A *mixed* equivalence class contain both stød-bearing and stød-less phones.

#### 5.3.2.2 Out-of-vocabulary words

The phonetic dictionary must cover the words in the training data completely, but we do not know of any ASR system with complete dictionary coverage with respect to input using standard word-based dictionaries and LMs. The vocabulary of languages constantly increases by adapting foreign words, generating named entities referenced by nouns or verbs[13], and language-internal evolution such as changed spelling or normalisation of slang also add to vocabulary growth.

Table 5.13 shows the OOV statistics for Stasjon03 and Stasjon06. The *coverage* is the proportion of unique words that occur in a test set and are represented in the ASR dictionary, i.e. 93.5% of the unique words in the Stasjon03 transcripts have phonetic representations in the phonetic dictionary and 6.5% do not. These words cannot be recognised and will decrease WER performance.

| | OOV | Types | Coverage | OOV tokens | Tokens | OOV rate |
|---|---|---|---|---|---|---|
| Stasjon06 | 369 | 5683 | 93.5% | 6995 | 79889 | 8.75% |
| Stasjon03 | 648 | 7311 | 91.1% | 735 | 112062 | 0.66% |

Table 5.13: OOV statistics for Stasjon03 and Stasjon06.

Based on the coverage, the OOV tokens in a test set can be counted and an OOV rate can be calculated

[12]The HMM states are implicitly modelled using an ordered set of pdf-ids.

[13]E.g. Google is a name, a brand, a noun and a verb, though the verb was commonly accepted at a later date.

which is equal to a lower bound on WER for that system/test set pair.[14] It is only a lower bound because false recognition of a word could, especially in Danish, be recognised as two separate words and such an error could cause other words to be falsely recognised.

### 5.3.3 Tuning

We have been using Stasjon03 as a development set throughout this chapter and we also use it to tune the *beam*, *lattice-beam* and *max-active-states* decoder parameters by sweeping a range of parameter values. When we sweep values of one parameter, the two other decoder parameters are fixed to a predefined value, but because these three decoder parameters influence each other, we might not find the optimal parameter settings without considering all parameters at the same time. Therefore, we use parameter sweeping to identify smaller parameter spaces to explore exhaustively.

An example of parameter sweeping on Stasjon03 can be seen in Figure 5.4. To create the graphs, we fixed the parameters at *beam*=15, *max-active-states*=8000 and *lattice-beam*=8 when they are not sweeped. The blue graph and left y-axis indicates the RTF performance and the WER performance is plotted with green.

The graphs indicate a smaller parameter space that can be searched more exhaustively. We then either repeat parameter sweeping with better fixed parameters or use grid search. In this case, we used near-optimal fixed values, but the *max-active-states* parameter was fixed at a sub-optimal value. We would repeat that parameter sweep and then conduct grid search on the new smaller parameter space.

If we were to choose a restricted parameter space from the graphs in Figure 5.4, a reasonable space to search would be e.g. *beam*=[13-15] because WER stops improving and *max-active-states*=[4000-7000] and *lattice-beam*=[7-8] which are the ranges with the best WER and RTF performance.

For each evaluation, the decoder produces a lattice that we rescore with the same LM but with different weights to arrive at an optimal LM weight at the same time. In Section 5.3.4, we reuse the LM weight and decoder parameters to decode Stasjon06.

Table 5.14 shows the WER evaluation in all experimental conditions on Stasjon03. We increased the beam width and lattice beam width until we achieved the best WER performance under the constraint that $RTF < 0.8$. For many systems, the WER improvement plateaued before the RTF threshold was reached.

Consistently, stød+pitch systems based on a triphone GMM AM significantly outperform their baselines and this is the case irrespective of whether the system is trained on MFCC or PLP features. There are no significant performance differences for mono0a systems.

[14] Because WER is an *error rate*, it should be as low as possible and the OOV rate is a bound on how good the performance in terms of WER can become.

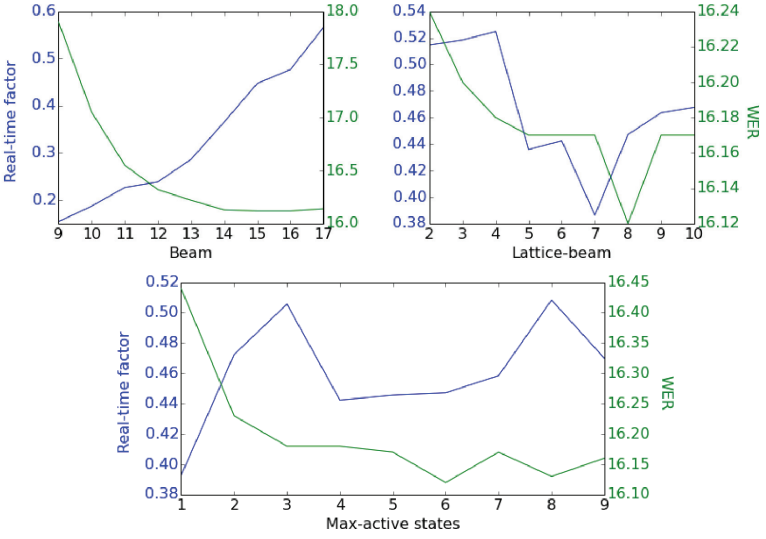



Figure 5.4: Parameter sweep on Stasjon03 for three decoder parameters. The x-axis in the bottom graph is in thousands.

We also observe highly significant WER performance improvement in the MFCC+ stød condition for tri4a, tri4b and nnet5c systems and the MFCC+stød nnet5c system outperforms all other systems in Table 5.14 including the MFCC+stød+pitch nnet5c system ($p < 0.001$). The performance increase is however not significant compared to the PLP+stød+pitch nnet5c system.

Another observation is that there is little increase or decrease in performance when we compare the WER of PLP and PLP+stød systems without pitch features. There is no discernible consistency to how performance is impacted by stød annotation.

### 5.3.4 Results

Table 5.15 show the WER evaluation on Stasjon06 using the decoder parameters and LM weight tuned on Stasjon03 for all systems in all conditions. In general, the WER performance of GMM-based stød+pitch systems that are trained on MFCC features – or LDA features derived from MFCCs – are lower than the performance of PLP+stød+pitch systems, but only significantly when we compare tri4b systems. When we

| System | PLP | | | MFCC | | | 1st vs. 2nd |
|---|---|---|---|---|---|---|---|
| | Baseline | Stød | Stød+ pitch | Baseline | Stød | Stød+ pitch | |
| mono0a | 54.04 | 54.06 | 53.66 | 52.99 | 52.81 | **52.57** | ~ |
| tri1 | 29.50 | 29.47 | 28.02*** | 28.90 | 29.12 | **27.96***** | ~ |
| tri2a | 27.37 | 27.35 | 25.88*** | 27.03 | 26.93 | **25.89***** | ~ |
| tri2b | 26.06 | 26.18 | 25.47*** | 26.06 | 25.83* | **25.09***** | *** |
| tri3b | 23.55 | 23.63 | 22.78*** | 23.78 | 23.51** | **22.57***** | * |
| tri4a | 20.66 | 20.65 | 19.99*** | 20.88 | 20.52*** | **19.81***** | * |
| tri4b | 19.77 | 19.70 | 19.17*** | 19.94 | 19.60*** | **18.94***** | ** |
| nnet5c | 15.89 | 15.81 | 15.48*** | 15.50 | **15.36***** | 15.69 | ~ |

Table 5.14: %WER comparison on Stasjon03 for baseline, stød-informed and stød+pitch systems. The performance of the best performing system is in blue and bold-faced. Statistical significance over the system in the column to the left is denoted by symbols: ~ if $p > 0.05$, * if $p < 0.05$, ** if $p < 0.01$ and *** if $p < 0.001$. For instance, the MFCC+stød+pitch tri3b system is significantly better than the MFCC+stød tri3b system in the table. Blue asterisk denote significant performance increase to the next best system, e.g. the MFCC+stød+pitch tri4a system outperforms the PLP+stød+pitch tri4a system at significance level *.

compare PLP+stød and MFCC+stød conditions, there is not a clear performance difference across systems, sometimes the PLP+stød system performs better than corresponding MFCC+stød system and vice versa. We will compare experimental conditions more in-depth below.

#### 5.3.4.1 Baseline vs. explicit stød modelling

The monophone (mono0a) MFCC+stød system improves significantly over the baseline and outperforms the PLP+stød mono0a system ($p = 0.006$). There is no significant difference between baselines or between MFCC+stød and PLP+stød for tri1 and tri2a systems, but the MFCC+stød tri2b system performs significantly better than the baseline and PLP+stød ($p = 0.016$). We also see significant improvement over the baseline for MFCC+stød tri3b and tri4b systems ($p < 0.001$ and $p = 0.001$) and for the PLP+stød tri4b system ($p = 0.024$). The MFCC+stød nnet5c system significantly outperforms both the MFCC baseline at $p = 0.004$ and the PLP+stød condition at $p = 0.001$.

| System | PLP | | | MFCC | | | 1st vs. 2nd |
|---|---|---|---|---|---|---|---|
| | Baseline | Stød | Stød+ pitch | Baseline | Stød | Stød+ pitch | |
| mono0a | 48.86 | 48.66 | 48.77 | 47.49 | **47.05**** | 47.30 | ~ |
| tri1 | 26.13 | 26.12 | 24.61*** | 25.77 | 25.85 | **24.38***** | ~ |
| tri2a | 24.01 | 23.85 | 22.66*** | 23.95 | 23.99 | **22.48***** | ~ |
| tri2b | 23.20 | 23.06 | 22.22*** | 22.72 | 22.42* | **22.16** | ~ |
| tri3b | 20.05 | 19.94 | 19.29*** | 20.37 | 19.95*** | **19.22***** | ~ |
| tri4a | 17.61 | 17.55 | 17.07*** | 17.72 | 17.54 | **16.88***** | ~ |
| tri4b | 16.85 | 16.64* | 16.49 | 17.14 | 16.81** | **16.17***** | ** |
| nnet5c | 13.50 | 13.33 | 13.17 | 13.28 | **13.08**** | 13.38 | ~ |

Table 5.15: %WER comparison on Stasjon06 for baseline, stød-informed and stød+pitch systems. The best performing system/experimental condition is in blue and bold-faced. Statistical significance over the system in the column to the left is denoted by asterisk: ~ if $p \geq 0.05$, * if $p < 0.05$, ** if $p < 0.01$ and *** if $p < 0.001$.

#### 5.3.4.2 Explicit stød modeling and pitch-related features

The performance of the MFCC+stød mono0a system is significantly better than the PLP+stød+pitch system, but not the MFCC+stød+pitch mono0a system, which scored second best in the evaluation.

The MFCC+stød+pitch and PLP+stød+pitch tri1 and tri2a systems perform similarly according to WER and improve significantly over other experimental conditions ($p < 0.001$).

The PLP+stød+pitch tri2b system significantly outperform PLP+stød ($p = 0.005$) but the MFCC+stød+pitch tri2b does not significantly outperform MFCC+stød.

We see highly significant performance improvements for the stød+pitch tri3b and stød+pitch tri4a systems at $p < 0.001$ and the MFCC+stød+pitch tri4b also very significantly outperforms MFCC+stød as well as all other conditions ($p = 0.002$). PLP+stød+pitch does not outperform the PLP+stød condition, but does significantly improve over MFCC+stød.

While the MFCC+stød condition still outperforms other nnet5c systems, the performance increase is not significant compared to PLP+stød+pitch, but is significant compared to MFCC+stød+pitch at $p = 0.006$.

#### 5.3.4.3 Real-time performance

To complete the system comparison, we show the average RTF in Table 5.16. The RTF is an average of speaker-specific RTF averages. Many RTF averages are well under the 0.8 threshold, but the RTF performance is not very different from the average RTF on Stasjon03.

| | PLP | | | MFCC | | |
|---|---|---|---|---|---|---|
| System | Baseline | Stød | Stød+ pitch | Baseline | Stød | Stød+ pitch |
| mono0a | 0.77623 | 0.811311 | 0.690785 | 0.782925 | 0.793596 | **0.663294** |
| tri1 | 0.671715 | 0.703715 | 0.552723 | 0.67196 | 0.650815 | **0.535484** |
| tri2a | 0.678857 | 0.713236 | 0.535778 | 0.652732 | 0.666068 | **0.534091** |
| tri2b | 0.428707 | 0.470557 | 0.421551 | **0.382892** | 0.391363 | 0.39765 |
| tri3b | 0.317045 | 0.327989 | 0.316125 | **0.29037** | 0.315387 | 0.291656 |
| tri4a | 0.470596 | 0.484545 | 0.449796 | 0.436046 | 0.457284 | **0.420969** |
| tri4b | 0.541336 | 0.568017 | 0.521224 | **0.484218** | 0.51061 | 0.488485 |
| nnet5c | 0.784921 | 0.741404 | **0.695695** | 0.734551 | 0.718506 | 0.721416 |

Table 5.16: Average Real-Time Factor evaluation for decoding on Stasjon06. The best performance for a system/experimental condition is in blue.

When comparing baseline MFCC vs. PLP and MFCC+stød vs. PLP+stød etc., MFCC-based systems are faster than PLP-based systems. The only exceptions are the baseline tri1 systems and stød-informed nnet5c systems.

To further measure RTF improvement, we perform a paired difference t-test on per-speaker average RTF across all systems. For mono0a systems, there is a significant RTF improvement when adding pitch information (at $p = 0.019$ for PLP+stød+pitch and at $p = 0.001$ for MFCC+stød+pitch) but no significant difference between the two stød+pitch conditions. It is the same for tri1, i.e. significant speed-up over MFCC+stød and PLP+stød ($p < 0.001$ and $p = 0.0019$, respectively) and tri2a ($p < 0.001$ in both conditions), but no significant difference between pitch-based systems.

For tri2b systems, there is no significant difference between MFCC-based systems. The difference between the PLP baseline and PLP+stød+pitch is not significant, but the PLP+stød system is significantly slower ($p = 0.0067$) and PLP-based systems are generally slower than MFCC-based systems: the slowest MFCC-based system is significantly faster than the fastest PLP-based system ($p = 0.0038$).

The MFCC+stød tri3b system is significantly slower than other MFCC-based tri3b systems, but the difference between the MFCC baseline and MFCC+stød+pitch is not significant.

The tri4a MFCC+stød+pitch systems is significantly faster than all other tri4a systems, with the exception of the MFCC baseline. The PLP+stød+pitch system is significantly faster than both baseline and PLP+stød ($p = 0.025$).

There is no significant difference between MFCC-based tri4b systems and the tri4b PLP+stød+pitch system is significantly faster than the MFCC+stød system, but not the PLP baseline.

The PLP+stød+pitch nnet5c system is significantly faster than other PLP-based nnet5c systems ($p < 0.001$) while there is no significant change betwen MFCC+pitch+
stød and MFCC+stød systems.

### 5.3.5 Analysis

#### 5.3.5.1 Effects of stød modeling and pitch-related features

The results show that modelling stød in the phonetic dictionary gives significant improvement in WER for systems based on LDA-transformed MFCC-features. The improvement in WER comes at the expense of RTF performance for GMM-based systems as illustrated for PLP-based tri2b systems in Figure 5.5. Generally, GMM-based systems that explicitly model stød have a higher RTF than the corresponding baseline, while DNN-based systems achieve a lower RTF. Adding pitch-related features tend to compensate for the increase in RTF or even speed up decoding and improve WER except in one case.

In WER evaluation, the MFCC+stød nnet5c system outperforms all other systems and experimental conditions and only the PLP+stød+pitch system is faster. The most advanced GMM-based systems (tri4b) perform significantly better than their respective baselines when stød is explicitly modelled ($p = 0.001$ and $p = 0.024$ for MFCC+stød and PLP+stød, respectively).

The GMM-based systems that use LDA-projected features (tri2b, tri3b, tri4a and tri4b) all outperform their baseline in terms of WER. The performance increase is significant for MFCC-based tri2b, tri3b and tri4b systems at $p \leq 0.016$ and significant for the PLP-based tri4b system ($p = 0.24$), but not for tri4a system. LDA projection may be important because it is preceded by feature splicing, which concatenates features such that they represent a longer time window before (See Section 2.5.1 for description of LDA.).

The three systems tri2a, tri2b and tri3b use the same number of Gaussians and tied states and differ only in the feature type used – $\Delta$+$\Delta\Delta$ features, LDA-transformed features and LDA and SAT-transformed features (See Table 5.8). For the PLP-based systems, we see no correlation between the feature type and the significance of WER improvement over the baseline, but we see that the WER performance improvement

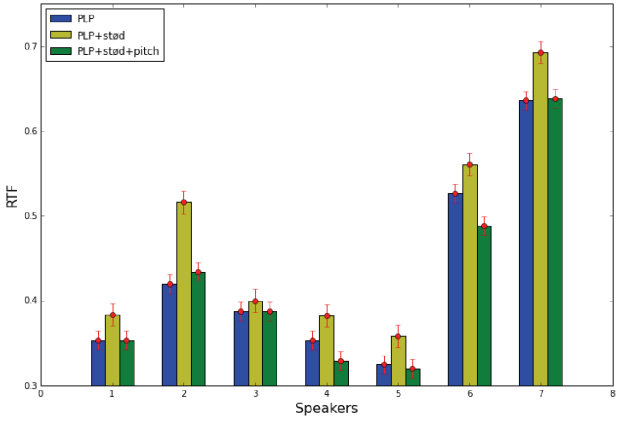



Figure 5.5: The impact of modelling stød in the phonetic dictionary and adding pitch-related features on the real-time factor. Adding stød increases the factor, but adding pitch-related features compensates.

between the MFCC baseline and MFCC+stød conditions goes from not significant (tri2a) to significant (tri2b) and to highly significant (tri3b) with the change in feature type.

The correlation between WER improvement can be observed in Tables 5.14 and 5.15 and suggests that features which represent a wide acoustic context are better at modelling stød.

### GMM AM complexity

Table 5.17 gives an indication of the change in the AM as a result of adding stød in tri4b systems.

When adding stød, the number of transition-states increase by approximately 10000 states and 22000-25000 more transitions need to be trained, while the number of Gaussians remain almost the same. The amount of estimated probability distribution functions (pdf) remains stable across baseline and stød-informed systems and is not impacted by the increase in transition-states and transition-ids. So stød increases the complexity of the AM and the number of transitions, but does not increase the descriptive power of the AM in terms of estimated pdfs.

Though the model becomes more complex, the increase in performance from the MFCC baseline to MFCC+stød is significant at $p = 0.001$. The increase from the PLP baseline to PLP+stød is significant at $p = 0.0241$ and supports the conjecture that explicit stød modeling in the PM is beneficial in Danish ASR.

| tri4b | Phones | PDFs | Transition-states | Transition-ids | Gaussians |
|---|---|---|---|---|---|
| PLP | 325 | 3834 | 31965 | 63970 | 60102 |
| MFCC | 325 | 3834 | 31555 | 63150 | 60097 |
| PLP+stød | 453 | 3848 | 44182 | 88404 | 60088 |
| MFCC+stød | 453 | 3761 | 43039 | 86118 | 60113 |

Table 5.17: AM statistics for tri4b systems. When estimating the PDT, the maximum number of leaves was specified as 4800 and max number of Gaussians was 60000. Phones refer to the number of word-position dependent phones and include 5 silence phones.

**Stød independence**

We have now determined that stød has an impact on the AM, but whether the AM actually models stød-bearing phones separately has not been confirmed. If the AM models stød-bearing phones separately, we can observe this in system-specific equivalence classes.

For the tri4b MFCC+stød system, there are 165 equivalence classes of word-position dependent phones out of which 59 contain stød-bearing phones. 43 are independent equivalence classes such as `['?e_B, ?e_B]`, `['?y_I, ?y_I]` or `[?o_E, '?o_E]` which are word-position dependent phones that we have forced to become phone aliases. Some independent equivalence classes contain phones from different word positions such as `['?y_B, '?y_S, ?y_S, ?y_B]` and `[?A_E, ?A_I, '?A_I, '?A_E, ?A_S, '?A_S]`. Table 5.18 shows the statistics of all stød-informed systems and the independent and mixed equivalence classes can be seen in Appendix B.4.

| Experimental condition | Classes | Independent | Mixed |
|---|---|---|---|
| PLP+stød | 167 | 45 | 15 |
| MFCC+stød | 165 | 43 | 16 |
| PLP+stød+pitch | 151 | 34 | 24 |
| MFCC+stød+pitch | 171 | 43 | 19 |

Table 5.18: Stød equivalence classes for tri4b systems. Independent classes contain only stød-bearing phones and mixed classes contain both stød-less and stød-bearing phones. All phones are word-position dependent and silence phones are not included.

Irrespective of the experimental condition, the number of independent equivalence classes outnumber the mixed classes. The PLP+stød+pitch system has fewer independent classes and more mixed classes out

of fewer total classes, but also features much larger equivalence classes, e.g. `[?W+_S, ?W_E, '?W_S, ?W+_E, '?W_E, '?W+_S, '?W+_E, ?W_S]`. Appendix B.4 also shows the equivalence classes for nnet5c systems where only the MFCC+stød nnet5c system differs by having an extra mixed class (`[?m_E, m_E]`).

Because independent classes contain phones from different word positions, merging word-position dependent phones decrease the likelihood of the data less than merging stød-bearing and stød-less phones in some cases which indicates that the distinction between stød-bearing and stød-less phones is sometimes more important than word-position.

#### 5.3.5.2 Recognition Errors

The top 10 confusion pairs, substitutions, deletions and insertions for MFCC+stød and MFCC+stød+pitch nnet5c systems evaluated on Stasjon06 are displayed in Appendix B.5. For both systems, the common recognition errors are small function words which is common in large-vocabulary ASR systems. The top 6 confusion pairs are phonetically similar such as [de/di], [u/o], [i/e] and [Ob@-n/Obn].

The most common recognition error is deletion, insertion or substitution of the word punktum (EN: period). punktum is not part of any frequent confusion pair, but is included in 337 low-frequent confusion pairs which are displayed in Appendix B.5.3. The words often confused with punktum bear no phonetic resemblance to punktum or each other and can be both noun, verb, function word, named entity etc. No pattern is discernible from the confusion pairs, but manual investigation of the alignment suggests the problem is inconsistent transcription in the training and test data.

The text preprocessing converts sentence final punctuation to spoken form because punctuation is usually dictated, however the dictation turns out to be inconsistent, i.e. there is a period at the end of a sentence whether it is spoken aloud or not, and frequently recognition errors such as the one in Figure 5.6 can be seen in the evaluation.

```
id: (46-r6110007-379)
Scores: (#C #S #D #I) 6 0 1 0
REF:  arrangér alle markerede afsnit efter længde PUNKTUM
HYP:  arrangér alle markerede afsnit efter længde *******
Eval:                                              D
```

Figure 5.6: Recognition error report from sctk featuring punktum (EN: Sort all high-lighted paragraphs by length). Capitalised words are erroneous and the error type D stands for deletion.

The ASR output (HYP) does not end in punktum because it is not spoken in the audio. Unfortunately, the reference contains the word and a deletion is counted towards the final WER. This would explain the high

number of errors, the inconsistent pattern in the confusion pairs in Appendix B.5.3 and the error occurring across ASR systems with different feature and stød combinations.

The reason for the large number of insertions seems to occur primarily when the ASR system tries to decode named entities or repetition utterances. Named entities are e.g. names of towns or first name and surname without utterance-final period and repetition utterances are the same word repeated three times. If the decoding fails as in the second and fourth example in Appendix B.5.3.2, the LM frequently inserts punktum before predicting the end of the utterance. We conjecture that many occurrences of sentence-final punktum in the LM training data leads to over-generation.

This problem seems to be specific to the data set. We use the transcripts from the training data to estimate the language model rather than a much larger newswire corpus because it fits the domain which includes utterances that consist only of repetitions and named entities. However, the transcription does not always faithfully reflect the utterance and the text preprocessing cannot take this into account.

## 5.4 Discussion

The presented results show that modelling stød improves WER performance in most pairwise system comparisons. If we compare the WER performance of the PLP baseline and PLP+stød conditions and the PLP+stød to PLP+stød+pitch conditions in Table 5.15 system by system, there is a consistent improvement in WER.

If we compare the MFCC baseline to MFCC+stød and MFCC+stød to MFCC+stød+pitch system by system, the improvement is less consistent, but many MFCC-based systems outperform PLP-based systems. The experiments indicate that WER improvement can be gained by explicitly modelling stød without increasing the number of Gaussians, but also that acoustic features which correlate with stød can further exploit stød annotation.

When we compare nnet5c systems, there is consistent WER improvement between the PLP baseline and PLP+stød, and PLP+stød and PLP+stød+pitch and a significant improvement from the PLP baseline to PLP+stød+pitch ($p = 0.002$). MFCC+stød significantly outperform all other DNN systems except PLP+stød+pitch.

### 5.4.1 Stød annotation

The analyses above indicate that explicit modeling of stød in the phonetic dictionary is relevant and is robust in the sense that data-driven methods do not cluster stød-bearing phones with stød-less phones. The decision tree algorithm uses the manually generated splitting criteria to separate stød-less and stød-bearing

variants of the same phone, i.e. the likelihood of the data increases, when we model stød. State-tying can cluster stød-bearing and stød-less phones, but the analysis shows that this happens only in some cases and that the likelihood of the data decreases less if word-position dependent-phones are clustered than stød-bearing and stød-less phones. Also several stød-bearing word-position dependent phones are not clustered with another phone.

The observations above hold across all experimental conditions for the most advanced GMM-based systems (tri4b) and DNN-based systems (nnet5c) and suggest that stød is sometimes more important than word-position and robust to clustering.

For this fairly simple task (compared to (semi-)spontaneous speech in medical dictation), automatic phonetic transcription has proven sufficient for pronunciation modelling. A phonetic dictionary specifically designed for Språkbanken was made available in 2015, but it was not possible to incorporate the new data into the experiments in the thesis. eSpeak is rule-based and while the generated dictionary provide good results, there are no descriptions of the principles and theory used, the affixation of diacritic symbols is unordered and some transcriptions are likely to be of questionable quality. The liquidated company from which all the Språkbanken speech data comes, Nordisk Språkteknologi, reportedly placed substantial time and effort into validation and quality assurance of their linguistic data (Andersen, 2008), but this claim is challenged by the inconsistent transcriptions and corrupted data found during the development of this recipe. Despite this, it is important that the dictionary is incorporated into the sprakbanken recipe at some point because it is a high quality manually created phonetic dictionary with stød annotation that can potentially give significant improvements in performance.

Stress is not modelled in the ASR systems presented in this chapter. In his enumeration of significant conditions for stød, Hansen (2015) notes that stød primarily (with few exceptions) occur on syllables with primary or secondary stress. He also notes that this distribution of stød is phonologically-based and the co-occurrence is not necessarily maintained in (semi-)spontaneous spoken language or the transcriptions produced by eSpeak. Hansen's own observation support the co-occurrence, but whether his observation is general or primarily pertains to his own data set is unknown. The impact from adding stress annotation and stød might not be complimentary if they primarily occur together.

### 5.4.2 Language model

The IRSTLM toolkit was used to estimate the LM. It is possible that a different LM toolkit or recurrent neural network LM rescoring can lead to significant performance improvements, but it is beyond the scope of this thesis. The LM is trained on transcripts from the training data and models general read-aloud speech but also include commands, number series and spelling which are specific to the test set.

We do not believe that high WER performance on the Stasjon06 (or Stasjon03) indicate how well the ASR system can be used for e.g. medical dictation because of the high proportion of repetition utterances, single named entity utterances and spelling aloud utterances which do not reflect the dictation task. Dictation as a speech genre is spoken language intended for written documentation and to have a more accurate indication, a different test set should be used for evaluation or only sentence-like utterances from Stasjon06 should be used for evaluation. Repetition, spelling and single named entity utterances can be used to evaluate according to e.g. phone error rate to give an intrinsic measure of the AM performance.

#### 5.4.2.1 Dictionary size

The LM and phonetic dictionary must correspond because it is not possible to recognise a word in the LM if it is not in the dictionary, so the dictionary size and coverage is important for performance. A 65000 word dictionary is considered large vocabulary in English (Adda-Decker & Adda, 2000)[15], but for languages like German and Danish with productive compounding, dictionaries need to cover a larger vocabulary to achieve similar word coverage. Adda-Decker & Adda (2000) compare lexical coverage across English, Japanese, Italian, French and German. The comparison can be seen in Table 5.19 where we have added a similar analysis for Danish based on the unigram frequency list from the NST n-gram frequency lists we used in Section 5.2.4. To achieve an OOV rate of 1%, a vocabulary size of ca. 290000 of the most frequent words is required which is 4.46 times larger than vocabularies for corpora in English, Italian, French and Japanese.

| Language | German | English | Italian | French | Japanese | Danish |
|---|---|---|---|---|---|---|
| Corpus | Frankfurter Rundschau | WSJ | Sole24 | LeMonde | Nikkei | Språkbanken |
| Tokens | 36M | 37.2M | 25.7M | 37.7M | 180M | 290M |
| Types | 650K | 165K | 200K | 280K | 623K | 2.8M |
| 5k coverage% | 82.9 | 90.6 | 88.3 | 85.2 | 88.0 | 80.58 |
| 20k coverage% | 90.0 | 97.5 | 96.3 | 94.7 | 96.2 | 90.45 |
| 65k coverage% | 95.1 | 99.6 | 99.0 | 98.3 | 99.2 | 95.78 |
| 65k-OOV% | 4.9 | 0.4 | 1.0 | 1.7 | 0.8 | 4.22 |

Table 5.19: Cross-lingual comparison of lexical coverage. The analysis of Danish lexical coverage is based on the NST unigram frequency list and the remaining numbers are from Adda-Decker & Adda (2000). 5k coverage% of 82.9 means the most frequent 5000 types cover 82.9% of all tokens in a corpus.

[15]This may have increased in recent years as it is a moving target.

The phonetic dictionary used in sprakbanken contains 65667 entries and model all unigrams in the training data and cover most of Stasjon03. Considering that 91.25% (8.75% OOV rate) is a low dictionary coverage compared to the coverage in Table 5.19, the evaluation does not suggest that the LM is poorly estimated on too little data. The LM performs well on text genres that are similar to the training data such as Stasjon06 and Stasjon03, but we expect that it will perform worse on e.g. Parole48 or DanPASS-mono, where Parole48 will contain many OOVs words and the syntactic structure of spontaneous speech in DanPASS-mono is different from written text.

There are different strategies to handle poor generalisation, such as increasing the dictionary size by adding more data. We could add the unigrams from the NST n-gram database, the unused transcripts in DK-Parole, the DanPASS dialogues or Danish Wikipedia to adapt the vocabulary (and LM) to other domains and reduce OOV rate.

The OOV rate can also be reduced by modelling compounds. Figure 5.7 shows how compounds can be wrongly recognised as two or more words. This not an infrequent problem but does not figure high in recognition errors, because different compounds are part of the recognition errors. Compound-boundary annotation is available in the phonetic dictionary from NST and another reason the additional NST resources should be added to the recogniser.

```
id: (46-r6110007-206)
Scores: (#C #S #D #I) 4 2 0 1
REF:  ********* ARRANGER UNDERDOKUMENTER til dette dokument punktum
HYP:  ARRANGERE UNDER     DOKUMENTER     til dette dokument punktum
Eval: I         S         S
```

Figure 5.7: Recognition error of compound: underdokumenter/under dokumenter (EN: sub-documents/under documents). Capitalised words are erroneous.

#### 5.4.2.2 Lexical context vs. acoustic modelling

A critical point in the assessment of stød in ASR is whether an improvement attributed to improved acoustic modelling can be achieved using a larger language model and is a parallel to the dilemma in Chapter 1: stød is not present in some Danish dialects which are still understandable by other Danes, because lexical context is sufficient for Danish spoken language understanding.

There are several phrases and words where intersentential lexical and semantic context cannot disambiguate a particular meaning, e.g. Ingen elsker bønner vs. Ingen elsker bønder. Only stød can disambiguate the semantics of these sentences bar discourse context or world knowledge.

These examples may not occur frequently but there are cases where stød modelling can provide important information that may not be modelled in a word n-gram LM. The results in this chapter indicate that modelling stød is not only relevant in border cases, but provide significant performance improvement and we believe this indicates that the information in lexical context and stød is complementary. Several models contribute to the accuracy of ASR and more accurate acoustic modelling can cause a shift in probability mass in the AM that in turn influences the probabilities evaluated by the decoder. The influence from stød may be decreased with a different language model, but modelling stød will have a positive effect.

### 5.4.3 Acoustic model

The constraint on PDT leaves and Gaussians are identical at each step in the bootstrapping process across all systems in the evaluation. The descriptive power of an AM is highly correlated to these parameters and increasing the threshold improves performance. We have shown that modelling stød explicitly and with pitch-related features significantly increase performance in most cases across 8 different system configurations and 4 experimental conditions (not including baseline).

Adding pitch-related features in ASR is not a novel idea, but has demonstrated valuable performance gains in other languages and now also for Danish. It remains to be seen whether adding features that correlate specifically with stød can be added to AMs without compromising overall performance. Features such as PDD and PDM which seem to improve phone discrimination show promise, but acoustic features which correlate well with stød might degrade the classification of stød-less phones.

Online ASR systems have not been evaluated. RTF has been reported to see if stød modelling decreases decoding speed so real-time medical dictation is not possible. As Table 5.16 indicates, online decoding is not compromised and we have controlled as well as possible the decoding from outside influences that could slow disk I/O, memory consumption or CPUs. The surprising observation is that while stød-informed systems have a higher RTF than baseline systems, pitch-related features compensate for the decreased speed. Adding stød adds to the number of transitions in the AM that needs to be evaluated in decoding (See Table 5.17) and we conjecture that this causes the decrease in RTF that we see in Table 5.16. Based on this, we further conjecture that a shift in probability mass caused by adding pitch-related features makes it easier for the decoder to choose a path through the lattice which improves RTF performance.

### 5.4.4 Application to stød detection

The current ASR can detect stød as a forced aligner and an interesting application is to increase the size of the data used in Chapter 4, but to do so it is necessary to model the inconsistency in stød manifestation. As mentioned in Chapter 2, stød is absent in some Danish dialects and stød can be omitted or dropped for

various reasons. Before we force align data to create more labelled data to create training data for stød detection experiments, we must model the fact that stød sometimes does not occur, i.e. lexical entries whose phonetic representation contain stød annotation should also have a stød-less phonetic representation, e.g. bønder should have two pronunciation variants: [b?WnV] and [bWnV]. If the variation in stød manifestation is not modelled, the newly labelled data will not be suitable for the task.

Additionally, the phone set should be considered. So far, automatically generated transcriptions have been sufficient, but frame-level experiments such as the feature selection and detection requires a high accuracy annotation that eSpeak cannot provide or may require yet more pronunciation variants. This will however increase confusability in the phonetic dictionary. Pronunciation variants does not always translate into improvement in WER performance.

Pronunciation variants will however improve pronunciation modelling and we conjecture that a phonetic dictionary with pronunciation variants can be used to improve training. The first task in ASR training is forced alignment where the algorithm creates a WFST where the most likely path is used to align the data (Lu et al., 2013). The estimated alignment will be more accurate and a phonetic dictionary without pronunciation variants can be used in decoding.

### 5.4.5 Application to medical dictation

A data set containing medical dictation was made available by Mirsk. We intended to use the data for training purposes and extracted all recordings that were shorter than two minutes and their associated transcriptions – about 12 h. Unfortunately, the data set was inconsistent and in many cases, the transcripts did not belong to the aligned audio or the audio was shorter than the aligned transcript and vice versa. The audio files were in a variety of encodings and sample rates and we were not able to convert a large part of them to the required format because of a low sample rate. We tried to convert the sound into 8 kHz WAV format, but the audio files became corrupted or unintelligible because the header information information often did not specify the correct sample rate.

The transcripts contained many non-standardised abbreviations. A particular non-standard abbreviation variant was in a few cases specific to a transcriber, but we also observed intra-transcriber variation which make it difficult to handle in computer programs. A particular severe problem was the abbreviation for *Pulmonary stethoscopic exam/Cardiac stethoscopic exam* which is abbreviated steth. p et c according to Schroeder et al. (2003) but was written as stet p et c, stet p og c, stet c et p, stet c stet p and st pc etc. The abbreviation used does not reflect the pronunciation which also showed great variability and reduction because it is a standing phrase in the medical domain.

Ultimately, the data was too unstructured and noisy that we decided to discard it. Manually transcribing select data would take time from the experiments presented in this chapter and Chapter 6 which we believe are of more academic value. Instead, we will evaluate performance on the DanPASS-mono and Parole48 data sets to evaluate whether stød modelling can generalise to unseen data in Chapter 6. The two data sets pose problems that we would also encounter in medical dictation, e.g. spontaneous speech (DanPASS-mono), low dictionary coverage (Parole48) and different text domains.

### 5.4.6 The relation between pitch and stød

According to the statistical significance test, the performance increase from pitch-related features is more significant and consistent across GMM-based systems and feature types than the performance increase gained by modelling stød in the phonetic dictionary. Only the MFCC+stød+pitch nnet5c system is outperformed by the corresponding MFCC+stød system. The results do not directly indicate whether we could obtain better performance with a MFCC+pitch system, i.e. a system that does not annotate stød in the phonetic dictionary. We do not expect a MFCC+pitch tri4b or nnet5c system to reach the same level of performance as a corresponding MFCC+stød+pitch system, but we lack the experimental results to confirm the conjecture and that is a weakness of this study.

### 5.4.7 Chapter conclusions

The purpose of this chapter has been the development of baseline systems and implementation of stød-informed and stød+pitch systems to discover how stød can improve ASR using conventional ASR methods and feature sets. Conventionally in ASR experiments, the performance of the most advanced systems such as tri4b or nnet5c are reported. The 'over-reporting' across different feature types and AMs serve to document that performance improvement is consistent in stød-informed systems and serves to guide researchers who wish to reproduce the results in this chapter.

Repeating the experiments requires access to the same resources, tools and methodology. The sprakbanken setup is based on publicly available tools and resources and while the installation of the Kaldi toolkit is not trivial and considerable computing resources are necessary to train large scale ASR systems, the system can be used as a reference for further development and be used for both teaching purposes and hypothesis testing. Previously, such a system has not been available for Danish and our hope is that the availability can improve the state of speech technology research and education, which was ranked undesirably low in Pedersen et al. (2012).

The WER evaluations show a significant improvements when stød is modelled in the phonetic dictionary and AM in MFCC+stød systems. Adding pitch-related features consistently improve WER across PLP and

MFCC systems, and also compensate for an increase in RTF caused by a more complex AM. This is positive confirmation in support of the theory that stød as important for automatic speech recognition of Danish spoken language. It is also a positive result that adding features revealed to be salient for stød detection in Section 4.2.4 does not degrade WER or RTF. Performing additional experiments with acoustic features e.g. phase features and Peak Slope is a logical step in further research on stød modelling in ASR.

The Språkbanken repository includes similar, larger data collections for Swedish and Norwegian. These resources should also be made available as a recipe in Kaldi and a methodology already exist for Swedish, which will facilitate the development (Vanhainen & Salvi, 2014). While the language-specific data sets are quite substantial, the existence of several recipes will make it possible to combine systems across languages, share models and potentially research acoustic language identification. Crucially, experiments with DNN AMs and LMs, which are scale-dependent and require huge amounts of data to be trained appropriately, can be conducted.

A number of potential areas for future research based on the open recipe has been presented including:

- Modelling stress and other prosodic features
- LM experiments such as neural LMs, compound splitting and domain adaptation
- Multilingual and inter-lingual (Scandinavian?) speech recognition and model sharing

Compounding, model sharing and language ID ties in well with the current focus on ASR and NLP for closely-related languages in the last couple of years. The possibility to share methods, models and technology between languages and has been the focus of three workshops, i.e. LT4CloseLang and VarDial in 2014 and LT4VarDial in 2015. Accordingly to the Meta-Net whitepaper series, this research could be of interest to all Scandinavian countries (Pedersen et al., 2012).

# Chapter 6

# Augmenting stød-informed ASR with stød-related acoustic features

Explicitly modelling stød in the phonetic dictionary can lead to significant performance improvements. The experiments in Chapter 5 show a general improvement in word error rate (WER) while online decoding is still possible and indicate that stød carries valuable acoustic information that can be used to resolve lexical ambiguities.

The stability and persistence of independent equivalence classes in the acoustic model (AM) confirm that modelling stød improves the likelihood of the training data sufficiently that state-tying does not cluster stød-bearing phones with stød-less phones.

We also extended the MFCC and PLP feature vectors with pitch-related features and showed that WER performance could increase further and that the increased real time factor (RTF) in stød-informed systems was to some extent compensated when pitch-related features were added.

Given the above observaitions, a natural extension is to include the acoustic features that were found to correlate with stød in Chapter 4. To continue this line of research, we train GMM (tri4b) and DNN-based (nnet5c) ASR systems on MFCC vectors that are extended with voice quality features and evaluate these systems against ASR systems developed in the previous chapter.

## 6.1 Acoustic stød modelling

Log-pitch, probability-of-voicing and $\Delta$log-pitch were appended to the ASR features used in Chapter 5 to improve acoustic modelling and this showed significant improvement in several cases. Adding pitch-related features to an ASR system is not novel, the feature extraction was already implemented in the

toolkit, and the experiment has been conducted several times in the literature on other languages than Danish (Ghahremani et al., 2014; Riedhammer et al., 2013). Adding features that specifically correlate with stød is not guaranteed to improve performance, because stød is not very frequent and since it can be difficult to exploit voice quality features to improve ASR performance outside of a low-resource context (Fernandez et al., 2014). The Språkbanken data is limited in the terms of speech genre, but not limited not in terms of data size. We therefore expect it to be difficult to improve upon the MFCC+stød and MFCC+stød+pitch tri4b and nnet5c systems in Chapter 5.

Fernandez et al. (2014) conducted a series of experiments on Zulu and Lao where many of the features explored in Chapter 4 are included. The performance for Zulu and Lao are state-of-the art and included a feature not previously explored in this thesis. The Harmonic Richness Factor (HRF) will be included in the experiments in this chapter, so the feature sets we investigate are:

1. MFCC+log-pitch, probability-of-voicing and Δlog-pitch
2. MFCC+log-pitch, probability-of-voicing and Δlog-pitch, and Peak slope
3. MFCC+log-pitch, probability-of-voicing and Δlog-pitch, PDD 10-13 and PDM 13-14
4. MFCC+log-pitch, probability-of-voicing and Δlog-pitch and HRF

where PDM and PDD are abbreviations for Phase Distortion Mean and Phase Distortion Deviation.

We extend MFCC feature vectors because the best tri4b and nnet5c systems from Chapter 5 are based on LDA features derived from MFCCs. We use pitch-related features because they improve performance significantly for tri4b systems and HRF is a pitch synchronous feature.

Fernandez et al. (2014) use different late integration approaches, but enumerate early integration and integration in GMM AMs in their discussion of future research and we therefore append voice quality features to MFCC+pitch vectors before MLLT/LDA projection.

### 6.1.1 Harmonic Richness Factor

The Harmonic Richness Factor (HRF) describes the amount of harmonic information (periodicity) in the speech signal. The computation of HRF bears resemblance to H1-H2. H1-H2 is the difference between the first two harmonics $H_1$ and $H_2$. To estimate HRF, compute

$$HRF = \frac{\sum_{i=2}^{K} H_i}{H_1}, H_i \in f, i = 1...K \tag{6.1}$$

where $H_i$ is the amplitude of the $i$th harmonic and $K$ is the number of harmonic peaks in the frequency range $f$.

$H_1$ is the peak closest to F0, i.e. defined identically to the formulation of H1-H2. Though HRF bears resemblance to H1-H2, both HRF and harmonics-to-noise ratio are measures of harmonic information vs. noise in speech where H1-H2 measures the relationship between the first and second harmonic only. In the experiments in Chapter 4, harmonics-to-noise ratio was not a salient feature and we believe the normalisation, where the undefined regions assigned a value of -200 by Praat are redefined as $\text{HNR}_{lowbound}$ (Eq. 4.1), may be a significant reason why the harmonics-to-noise ratio was not found to contribute salient information. We investigate HRF because it does not require similar normalisation as the harmonics-to-noise ratio and the feature can be extracted with Covarep, which extracted several salient features in Chapter 4.

Unlike the harmonics-to-noise ratio, HRF is a pitch-related feature, i.e. we must find F0 before we can estimate HRF, and this could be problematic if F0 is irregular. However, the experiments in Chapter 5 show that pitch-related features can have a positive influence on performance.

If HRF also turns out to be salient in the following experiments, it suggests that some periodicity is present to track F0 and that energies in high frequency bands contain information related to stød. Low F0 and relatively high H2 has been observed for stød-bearing phones in the literature (Hansen, 2015; Fischer-Jørgensen, 1989) and Fischer-Jørgensen (1989) also finds stronger harmonics in higher frequency bands.

According to Keating et al. (2014), creaky voice can be categorised according to several properties such as low F0, irregular F0, glottal constriction, the presence of subharmonics and noise levels, but none of the categories mention relatively stronger harmonic peaks. The changes at high frequency bands suggest that this is where we may find a more accurate phonetic description of stød than '*stød is more than creak*'.

## 6.2 Method

On the entire Språkbanken corpus, we extract HRF, Peak Slope, PDM and PDD using feature extraction settings that are identical to those used in Chapter 4, (See Appendix B.3). We extend MFCC+pitch feature vectors with these features (only PDM 13-14 and PDD 10-13, not all 38 phase features), normalise the extended vector using mean subtraction and variance normalisation in Kaldi and train AMs using the methodology described in Chapter 5. We use early integration and add the new features before LDA projection, just as we did with pitch-related featues.

Performance is evaluated on Stasjon06, DanPASS-mono and Parole48 data sets and OOV statistics for DanPASS-mono and Parole48 can be seen in Table 6.1

| | OOV | Types | Coverage | OOV tokens | Tokens | OOV rate |
|---|---|---|---|---|---|---|
| Parole48 | 459 | 2473 | 81.4% | 1012 | 6935 | 14.6% |
| DanPASS-mono | 172 | 1141 | 84.9% | 5232 | 21418 | 24.4% |
| Stasjon06 | 369 | 5683 | 93.5% | 6995 | 79889 | 8.75% |

Table 6.1: OOV statistics for DanPASS-mono, Parole48 and Stasjon06.

The OOV rate is much higher in Parole48 and DanPASS-mono than in Stasjon06 and we expect to observe degraded performance on Parole48 and DanPASS-mono because of the OOV rate and the shift in text genre. The DK-Parole corpus is based on articles from newspapers and periodicals which regularly introduce new vocabulary when describing recent events, foreign places or persons, inventions and other named entities. The articles describe different topics and therefore the phonetic dictionary has the lowest coverage on Parole48, but because a specific OOV word does not occur frequently, the OOV rate is relatively low compared to DanPASS-mono. The DanPASS corpus also contain several named entities that are not in Språkbanken, such as places in the map or shapes in the geometric network and house described in DanPASS-mono. The phonetic dictionary has higher coverage, but the same named entities are described by all speakers and because they are high-frequent, the OOV rate is higher in DanPASS-mono than Parole48.

The syntax will also change because neither DanPASS-mono or Parole48 contain spoken commands, repetitions, spellings or name utterances, but are closer in structure to the text genre used in medical dictation.

### 6.2.1 Evaluation

We use the same training parameters as in Chapter 5 to train ASR systems on extended acoustic feature vectors. However, because we wish to discover if additional acoustic features can improve ASR, we optimise the LM weight, beam width, lattice-beam and max-active-states decoder parameters of using MFCC+stød+pitch systems for performance on Stasjon06, Parole48 and DanPASS-mono under the $RTF < 1$ constraint. We then use these parameters to decode the same test sets with tri4b and nnet5c systems trained on extended features vectors.

The motivation is that it is difficult to achieve a performance increase with novel acoustic features outside of limited-resource context and because we use Språkbanken for training, we are not limited by data resources. If we observe improved performance using novel acoustic features over a baseline that is optimised for our test sets, we believe it is the best indication of the utility of the novel features.

We use WER as the main performance metric and calculate statistical significance to compare system performance with MAPSSWE, similar to Section 5.3.2. We also report RTF performance and analyse the impact of the extended feature sets on performance using beam sweeps in Section 6.4.1.

We include the best tri4b and nnet5c system according to WER performance from Chapter 5 as baselines in the evaluation, i.e. the MFCC+stød+pitch tri4b system and MFCC+stød nnet5c system. Because the systems in this chapter all use a phonetic dictionary with stød annotation, we drop the 'stød' identifier and denote the two systems as the MFCC+pitch tri4b system and the MFCC nnet5c system.

In terms of hardware, we use a reserved server identical to the one used in the previous chapter to train the ASR systems on extended feature vectors and conduct the experiments.

## 6.3 Results

For each test set, the decoder parameters are tuned for performance with MFCC+pitch features and the best performance under the $RTF < 1$ constraint is chosen to evaluate the systems. The evaluation in Table 6.2 compares tri4b systems trained on MFCC+pitch+phase, MFCC+pitch+HRF and MFCC+pitch+Peak Slope to MFCC+
pitch features.

| Test set | Metric | Stasjon06 (16-7-8000) | Parole48 (15-8-8000) | DanPASS-mono (12-5-6000) |
|---|---|---|---|---|
| MFCC+pitch | WER | **16.12** | 33.53 | 56.41 |
| | RTF | 0.633 | 0.727 | 0.880 |
| MFCC+pitch+phase | WER | 16.25 | 33.68 | **55.96** |
| | RTF | 0.576 | 0.772 | 0.676 |
| MFCC+pitch+Peak Slope | WER | 16.73 | 33.83 | 57.38 |
| | RTF | **0.523** | **0.696** | **0.638** |
| MFCC+pitch+HRF | WER | 16.21 | **33.08** | 56.58 |
| | RTF | **0.523** | 0.708 | **0.637** |

Table 6.2: WER and RTF on Stasjon06, DanPASS-mono and Parole48 for all tri4b systems when decoder parameters are tuned for WER performance with MFCC+pitch features. The best WER performance for a test set/feature set is in blue , the best RTF performance is in red and both are bold-faced. No statistically significant performance improvements were measured using MAPSSWE.

On Stasjon06, the performance of MFCC+pitch+Peak Slope is significantly worse than all other systems

at ($p < 0.001$), but there is no statistically significant difference in performance between MFCC+pitch, MFCC+pitch+HRF and MFCC+pitch+phase. No system achieves significant improvements in WER on Parole48, but the WER performance on DanPASS-mono of the MFCC+pitch+Peak Slope system is again significantly worse than the other systems at $p < 0.001$ for MFCC+pitch+phase and $p \leq 0.042$ for MFCC+pitch and MFCC+pitch+HRF).

The MFCC+pitch tri4b system outperforms other systems on Stasjon06, but is outperformed by MFCC+pitch+phase on DanPASS-mono and by MFCC+pitch+HRF on Parole48. Also, the MFCC+pitch+Peak Slope and MFCC+pitch+HRF systems are faster as indicated by the lower RTF.

The same procedure is applied to evaluate nnet5c systems. We select MFCC+pitch, MFCC+pitch+HRF and MFCC+pitch+phase feature sets based on WER performance from Table 6.2 and train two nnet5c systems on MFCC+pitch+HRF and MFCC+pitch+phase feature sets. The evaluation is reported in Table 6.3.

| Test set | Metric | Stasjon06 (12-6-6000) | Parole48 (14-7-5000) | DanPASS-mono (12-5-6000) |
|---|---|---|---|---|
| MFCC | WER | 12.94 | **29.78** | 53.83 |
| | RTF | 0.704 | 1.035 | 0.871 |
| MFCC+pitch | WER | 13.10 | 30.38 | 54.73 |
| | RTF | 0.718 | 0.929 | 0.801 |
| MFCC+pitch+phase | WER | **12.16***** | 30.05 | **49.02*** |
| | RTF | **0.498** | **0.775** | **0.662** |
| MFCC+pitch+HRF | WER | 12.58*** | 30.38 | 51.06 |
| | RTF | 0.692 | 0.780 | 0.692 |

Table 6.3: WER and RTF on Stasjon06, DanPASS-mono and Parole48 for all nnet5c systems when decoder parameters are tuned for WER performance with MFCC+pitch features. The best WER performance for a test set/feature set is in blue, the best RTF performance is in red and both are bold-faced. Statistically significant WER improvement over the MFCC system is denoted by symbols: $\sim$ if $p > 0.05$, * if $p < 0.05$, ** if $p < 0.01$ and *** if $p < 0.001$.

The MFCC+pitch+phase system significantly outperforms other systems on Stasjon06 ($p < 0.001$) and DanPASS-mono ($p = 0.016$). The MFCC system achieves the best WER performance on Parole48, but this is not a significant improvement. The systems trained on extended acoustic features have lower RTF performance than systems trained on MFCC and MFCC+pitch features.

## 6.4 Analysis

### 6.4.1 Performance

In terms of tri4b performance, no clear picture emerges from the system evaluation in Table 6.2. Only the MFCC+pitch+phase tri4b system is significantly better than the MFCC+pitch+HRF tri4b system on DanPASS-mono, but otherwise different systems perform better on different test sets and none of the differences are significant. We do observe in all but one case that extended feature sets improve RTF performance. While increased speed in decoding is not the focus of these experiments, a potential improvement in WER performance may be indirectly achieved because the decoder parameters can be increased without compromising real-time capabilities.

The MFCC+pitch, MFCC+pitch+HRF and MFCC+pitch+phase tri4b systems significantly outperform the MFCC+pitch+Peak Slope tri4b system on Stasjon06 and DanPASS-mono, and based on this performance we train nnet5c systems on MFCC+pitch+HRF and MFCC+pitch+phase features sets. In the nnet5c evaluation in Table 6.3, the MFCC+pitch+phase nnet5c system significantly outperforms all other systems on DanPASS-mono and also on Stasjon06 in addition to having the lowest RTF across all evaluations. In terms of relative reduction, the MFCC nnet5c system achieves the best performance on Parole48 with a 2% reduction compared to the MFCC+pitch nnet5c system, MFCC+pitch+phase achieves more than 10% reduction on DanPASS-mono, and also 7.1% reduction on DanPASS-mono and Stasjon06.

To analyse the relationship between WER performance and RTF for the MFCC, MFCC+pitch, MFCC+pitch+HRF and MFCC+pitch+phase feature sets, we sweep the beam size of the nnet5c systems in Table 6.3 and plot WER and RTF on Stasjon06, Parole48 and DanPASS-mono to see if an additional performance increase can be gained. In Figures 6.1, 6.2 and 6.3, we abbreviate the feature set names as follows:

| Abbreviation | Full description |
|---|---|
| M | MFCC |
| MP | MFCC+pitch |
| MPH | MFCC+pitch+HRF |
| MPP | MFCC+pitch+phase |

Table 6.4: Abbreviation table for legends in Figures 6.1, 6.2 and 6.3.

The MFCC+pitch+phase and MFCC+pitch+HRF nnet5c systems are consistently faster by almost 0.18 RTF than the MFCC and MFCC+pitch nnet5c systems. The MFCC+pitch+phase system also consistently achieves a lower WER performance for all beam values in the parameter sweep. The MFCC+pitch+HRF

system also shows a constant improvement though at a smaller factor and the difference in WER performance of MFCC+pitch+phase over MFCC+pitch+HRF is significant at $p < 0.001$, which is visually apparent in Figure 6.1.

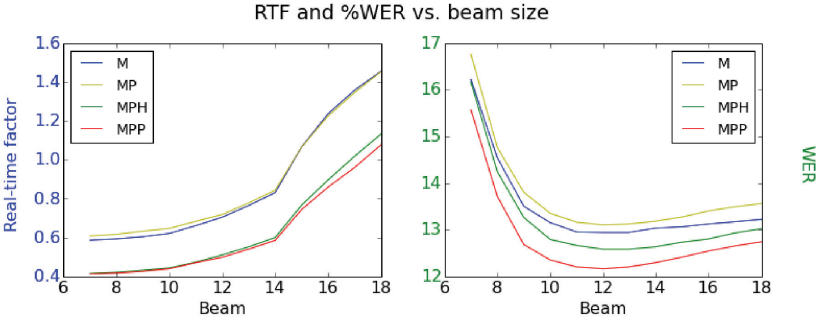



Figure 6.1: Beam parameter sweep on Stasjon06 for all feature sets. The best performance is achieved by MFCC+pitch+phase (MPP) with a beam size around 12.

The RTF performance gap between the MFCC and MFCC+pitch systems, and MFCC+pitch+HRF and MFCC+pitch+phase systems hold on the Parole48 test set in Figure 6.2. The MFCC+pitch system RTF does not degrade together with the MFCC system as in Figure 6.1 and this is the reason the MFCC system in Table 6.3 is slower than real-time. We also see that if we set the beam size to 14 for the MFCC+pitch+phase and MFCC+pitch+HRF systems, they can achieve 30.05% and 30.38% WER performance respectively and still decode in real-time[1]. The increased decoding speed can thus be translated into a WER improvement that closes the performance gap between MFCC+pitch and MFCC+pitch+HRF.

We also observe lower RTF for the MFCC+pitch+HRF and MFCC+pitch+phase systems in Figure 6.3, but the gap is smaller than on Parole48 and DanPASS-mono. The graph does not have a steep incline between beam sizes of 14 and 15 as in Figures 6.1 and 6.2, but shows a smooth trend where we can observe that the MFCC+pitch+HRF and MFCC+pitch+phase systems can use wider beam sizes than the MFCC and MFCC+pitch systems and still adhere to the $RTF < 1$ constraint.

To sum up the observations from Figures 6.1, 6.2 and 6.3, the WER performance for each feature set/test set with $RTF < 1$ is reported in Table 6.5. There is no change in WER performance on Stasjon06 because we already achieve best performance with beam size 12. The performance of the MFCC baseline on Parole48 decreases because we need to narrow the beam to obtain 0.950 RTF, but this does not lead

[1]If the beam is set to 15, then $RTF \leq 1.018$.

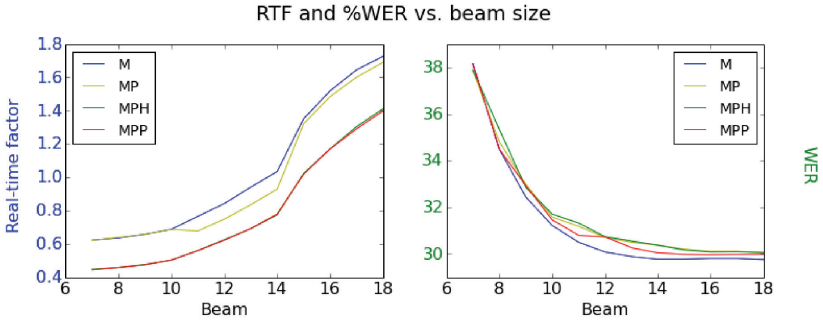



Figure 6.2: Beam parameter sweep on Parole48 for all feature sets. The best performance is achieved by the MFCC baseline (M) with beam size 12.

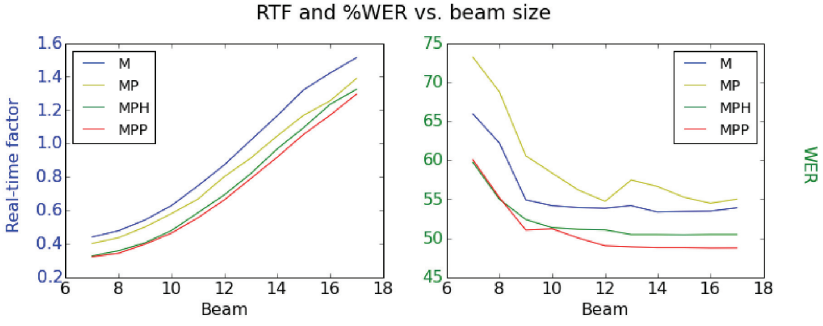



Figure 6.3: Beam parameter sweep on DanPASS-mono for all feature sets. The best performance is achieved by MFCC+pitch+phase (MPP) with a bea size of 12.

to significant change in WER and the baseline still achieves the best performance. The WER performance of the MFCC+pitch+HRF and MFCC+pitch+phase systems improve on DanPASS-mono, but only the MFCC+pitch+phase system improve significantly over Table 6.3.

| Test set | Metric | Stasjon06 | Parole48 | DanPASS-mono |
|---|---|---|---|---|
| MFCC | WER | 12.94 | **29.89** | 53.83 |
| | RTF | 0.704 | 0.940 | 0.871 |
| MFCC+pitch | WER | 13.10 | 30.38 | 54.73 |
| | RTF | 0.718 | 0.923 | 0.801 |
| MFCC+pitch+phase | WER | **12.16***** | 30.05 | **48.79***(**) |
| | RTF | 0.498 | 0.780 | 0.918 |
| MFCC+pitch+HRF | WER | 12.58*** | 30.38 | 50.46 |
| | RTF | 0.692 | 0.775 | 0.968 |

Table 6.5: WER and RTF on Stasjon06, DanPASS-mono and Parole48 for all nnet5c systems with the widest beam under $RTF < 1$. The best performance for a test set/feature set is in blue. Statistically significant WER improvement over the MFCC baseline is denoted by symbols: $\sim$ if $p > 0.05$, * if $p < 0.05$, ** if $p < 0.01$ and *** if $p < 0.001$. Asterisk in parenthesis denote improvement over the same feature set/test set in Table 6.3.

### 6.4.2 Stød independence

The independent and mixed equivalence classes that are clustered in state-tying during the training of the MFCC+pitch+phase and MFCC+pitch+HRF nnet5c systems can be seen in Appendix B.4 and Table 6.6 show the equivalence class statistics of all nnet5c systems used in Section 6.3.

Three observations about the shared equivalence classes:

1. All independent equivalence classes cluster phones by word position and stød
2. Two mixed classes likely contain errors
3. 8 out of 10 mixed classes cluster phones by word position

The two erroneous equivalence classes are `['d_B', '?d_B']` and `['d_E', 'd_I', '?d_S', 'd_S', '?d_E', '?d_I']` because [d] is a consonant and should by definition not be stød-bearing irrespective of whether it is a plosive or a stop and it turns out that the source of this error is in the the word *akkord* (EN: chord). Because this is an error, it is a positive outcome that state-tying consistently clusters the erroneous stød-bearing phones with their stød-less variants.

We can also see that all the shared independent equivalence classes exclusively contain stød-bearing vowels, that the alveolar fricatives [s] and [z] and the nasal [m] cluster by word position and that all the variants of [0] are in one cluster. When we inspect the phonetic dictionary, we count 413 word-internal

occurences of [?0] and [0] and two occurrences of word-initial [0], i.e. because only three variants are observed in the data, the unobserved and low-frequent variants are clustered with frequent variants. We conjecture that [?0] and [0] are clustered because the distinction does not increase the likelihood of the data sufficiently to resist state-tying.

| Feature set | Total | Independent | Mixed |
|---|---|---|---|
| MFCC | 165 | 43 | 16 |
| MFCC+pitch | 171 | 43 | 19 |
| MFCC+pitch+phase | 158 | 37 | 18 |
| MFCC+pitch+HRF | 166 | 45 | 13 |
| Shared | 38 | 28 | 10 |

Table 6.6: Equivalence class statistics for nnet5c systems and the number of mixed and independent equivalence classes that are identical across the nnet5c systems. Independent classes contain only stød-bearing phones and mixed classes contain both stød-less and stød-bearing phones. Note the large proportion of independent classes. All phones are word-position dependent and silence phones are not included.

## 6.5 Discussion

### 6.5.1 Extended feature sets

HRF and phase features show promise in Danish ASR and capture information in speech that improves accuracy and decoding speed. Adding Peak Slope to standard ASR features significantly degraded performance in tri4b system evaluation and we did not include it in the nnet5c systems evaluation.

The performance degradation we observe when we add the Peak Slope parameter is unexpected because the feature is ranked as salient in Chapter 4 and had a positive influence on WER performance in Fernandez et al. (2014). To learn why, we decoded Stajon06 with tri1, tri2a, tri2b, tri3b and tri4a systems trained on MFCC+pitch+phase, MFCC+pitch+HRF and MFCC+pitch+Peak Slope and plotted the performance of these systems in Figure 6.4. We also add the MFCC+stød and MFCC+stød+pitch systems from Chapter 5 in the comparison.

We clearly see that the MFCC+pitch+Peak Slope ASR systems do not improve at the same rate as systems that are extended with HRF or phase features when we apply LDA transformation. Phase features improve by 6% WER absolute which is nearly 3 times more improvement than we observe for

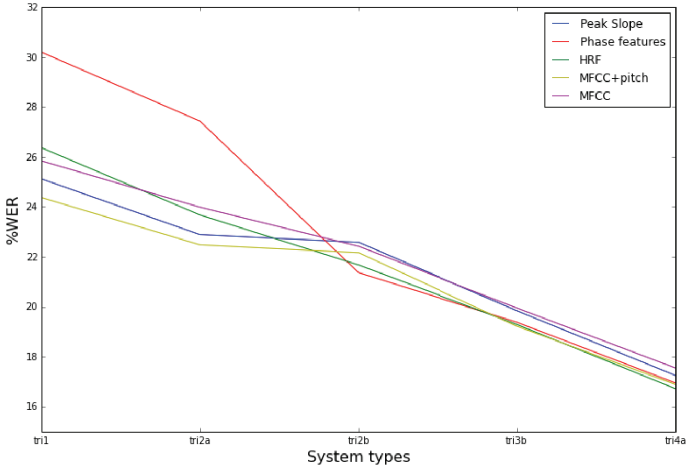



Figure 6.4: The impact of LDA projection on %WER performance in GMM-based ASR. The names identify the voice quality features that are appended to the MFCC+pitch feature vectors. The MFCC+stød and MFCC+stød+pitch systems refer to the systems in Table 5.15.

MFCC+pitch+HRF and MFCC+stød and 19 times ($\frac{6.08}{0.31}$) more than MFCC+pitch+Peak Slope (0.31 absolute improvement). We conjecture that a degradation was not observed in Fernandez et al. (2014) because 1) they extend PLP features with 10 features that include Peak Slope, HRF, H1-H2 etc. and 2) they use late integration and extend feature vectors after LDA projection.

The performance degradation we observe is therefore not an indication that Peak Slope is not a salient feature for the detection of stød, but suggests that Peak Slope is not suited for early integration.

### 6.5.2 Feature extraction speed

The phase feature extraction method used here is slower than real-time. The harmonics of the speech signal is computed using an adaptive harmonic model, which is an accurate, but slow estimation method. A faster estimation method such as peak picking could reduce the computation time of the harmonic analysis to approximately 10% at the expense of precision (Degottex & Stylianou, 2013). Matlab's `mex` functionality can speed up interpolation and compile programs, which should also speed up harmonic analysis by approximately 35%. Acceleration using either option makes real-time feature extraction possible.[2]

[2]Notes in the Covarep source code. We have not tested this claim.

The feature ranking experiment in Chapter 4 suggest that phase features are not only salient for stød detection, but for general phone discrimination as well. We do not know if the real-time/lower precision extraction method is sufficient to predict stød, but faster extraction is necessary if we want to use phase features in ASR. If real-time ASR is not required, e.g. in a post-editing scenario, phase features can still be used to improve ASR accuracy.

HRF can be estimated in real-time, but is sensitive to the quality of F0 estimation and pitch is difficult to estimate in noisy speech. HRF does not provide the same reduction in WER or RTF as phase features do, but is more straightforward to exploit because the estimation is fast and we consider HRF as a salient feature for the detection of stød and an addition to the features in select+.

### 6.5.3 Robustness

All corpora used in this thesis have been recorded in noiseless conditions or an office environment. We do not know how beneficial phase and HRF features are to performance at different levels of signal-to-noise ratios or reverberant noise. If the input is too noisy, the information contributed by HRF and phase features could become unreliable, especially in the case of HRF if the F0 estimate becomes less reliable.

Robustness in speech recognition is a very active research field that for instance use methods to add noise to sound files to simulate noisy environments. Using additive noise and room-impulse response, training and test data with different signal-to-noise ratios can be generated to test robustness of features and ASR systems. As future work, we propose a similar test to investigate the robustness of HRF and phase features because additive noise will have an impact on F0 estimation (Gerhard, 2003) and room-impulse responses can corrupt the estimation of the shape of the glottal pulse, which is correlated with phase features (Degottex & Erro, 2014).

### 6.5.4 Relevance to medical dictation

As demonstrated, modelling stød explicitly in the phonetic dictionary and using HRF and phase features increase performance when we use a narrow LM that fits the domain, which indicates that the performance increase is complementary to syntactic information. In medical dictation, it is difficult to obtain text data to estimate LMs, because medical data contains sensitive information and the ASR systems are forced to rely less on textual context. Exploiting stød is therefore of special interest in this and similar contexts because significant performance improvements can potentially be gained. Because of the slow phase feature extraction algorithm, HRF is more suited to augment the feature input to ASR systems, though it remains to be seen whether HRF is robust in noisy environments.

## 6.6 Chapter conclusions

On clean recordings in two different speech genres and three different data sets, the ASR systems augmented with phase features or HRF either achieve similar results or outperform the previously described stød-informed systems trained on MFCC or MFCC+pitch features. Though Peak Slope is an informative feature according to the experiments in Chapter 4, experiments show that the feature degrades performance in GMM-based ASR systems when we extend the acoustic feature vector prior to LDA projection.

A useful contribution from the added features is an increase in decoding speed, which makes it possible to traverse a larger part of the lattice. Although the best WER and RTF performance is gained using phase features, the slow feature extraction algorithm makes them impractical to implement and use in their current form. This can potentially be mitigated using faster algorithms for phase extraction that are based on peak picking, if the estimated features are shown to be sufficiently accurate.

On the basis of the results in this chapter, we consider HRF to be another feature that is salient to stød detection which supports the conjecture that stød is correlated with information in higher frequency bands.

# Chapter 7

# Summary and future work

The working definition of stød in this thesis is "stød is more than just creak". The present work does not attempt to answer what stød is in phonetic or acoustic terms, but investigates the assumption that stød is useful in Danish ASR. Our findings can be implemented in other speech recognisers by simply adding stød annotation to the phonetic dictionary and re-training the ASR system. The experiments in Chapter 5 demonstrate that significant improvements can be gained by explicitly modelling stød despite the increased phonetic alphabet. The analysis of equivalence classes confirm that stød is modelled separately and not re-clustered during state-tying.

Adding voice quality features such as phase features or HRF will require some development because HRF relies on F0 estimation and phase feature extraction is slower than real-time. The extraction algorithms must also align all the extracted features with MFCC or PLP features. We performed the alignment manually.

Because of variables concerning implementation, workflow choice (real-time ASR or ASR and post-editing), IT architecture etc., we cannot translate the improved WER performance to a theoretically or practically obtainable reduction in transcription time for a medical secretary. There is also no baseline study that we can base the calculation on. A user study could provide such an estimate, but is beyond the scope of this thesis and is left to industry or other researchers. However, the results in Chapters 5 and 6, and the research that led up to it have provided insights on the nature of stød.

## 7.1 Summary of experimental results

The experiments in the chapters of this thesis build on the research and conclusions in previous chapters and we summarise the findings of each chapter below.

### 7.1.1 Stød annotation

The study found that annotator agreement stemmed from agreement on a small set of highly frequent labels and because stød accounted for approximately 5% of the assigned labels, we focused the study on stød labels. The reliability study investigated both binary stød annotation and stød-bearing phones. Annotator agreement was challenged by the interpretation of the stød-bearing segment, which annotators could interpret as two vowels or as a long vowel, and the difference in interpretation created artificial disagreement in $\kappa$-scores. Correcting for this discrepancy and obvious errors resulted in high inter-annotator agreement and high annotator competence scores.

Based on the high agreement, we concluded that stød annotation is reliable when annotated by expert phoneticians and can form the basis for quantitative analysis, provided sufficient data can be obtained.

### 7.1.2 Stød detection

Based on the conclusions in Chapter 3 and additional stød-annotated data from the DanPASS and DK-Parole corpora, we estimate a number of statistical models to detect stød in Chapter 4. We extract 120 acoustic features and use a forest of randomised decision tree classifiers to rank the importance of the features for stød detection and ultimately choose 17 salient features.

The highest F1 score obtained by a classifier in binary stød detection was 0.17 on spontaneous speech and 0.32 on lab-recorded speech, and we find that stød detection is not possible when formulated as a binary classification task using the presented methods and data sets.

The conclusion is that identical performance can be achieved with logistic regression or linear support vector machine classifiers using less than 15% of the original 120 features and that the feature selection successfully identifies acoustic features that are that are salient for the detection of stød.

To investigate other methods to detect stød, we redefine stød detection as pairwise discrimination of stød-less and stød-bearing variants of the same phone. The average classification accuracy across all pairs ranges from near-perfect (0.92) in five-fold cross-validation to above average on unseen spontaneous speech (0.713, but with 0.266 variance). It is possible to detect stød if we reformulate the detection problem as pairwise discrimination.

We conclude that voice quality features such as Peak Slope, PDD and PDM are salient for stød detection, as well as pitch-related and standard ASR features. As far we have been able to ascertain, the correlation between phase features and stød is novel. The mentioned voice quality features are also sensitive to energy at higher frequency bands, whereas the features most commonly associated with stød such as

H1-H2 and harmonics-to-noise are not, which suggests that there is additional information to be found at higher frequencies that signals stød.

### 7.1.3 Stød in automatic speech recognition

The discrimination experiment show that information in PLP features can be used to discriminate between stød-bearing and stød-less samples, so stød can be implemented in ASR simply by annotating stød in the phonetic dictionary. We therefore want to study of how stød can be exploited in ASR using existing tools and methods.

There are two caveats: 1) the amount of training data needed to train ASR systems cannot be annotated for stød by experts, and 2) phonetic decision tree growth and state-tying can map stød-bearing phones to their stød-less counterparts essentially making one an alias of the other. Despite these differences to the experiments and data used in Chapters 3 and 4, we obtain significant improvements in WER performance in Chapter 5 for MFCC-based ASR systems, when stød is added to the phonetic dictionary in ASR systems. We observe additional highly significant WER improvements when we add pitch-related features, but not consistently in DNN AM-based systems.

Based on an analysis of the most advanced speaker-adapated GMM-based ASR systems trained on LDA-transformed features (tri4b), we confirm that only adding stød annotation to the phonetic dictionary increases the complexity of the acoustic model without increasing the descriptive power in terms of total number of Gaussians and probability density functions (pdf). The analysis of phonetic equivalence classes also show that stød is sufficiently informative that stød-bearing phones are modelled when the phonetic decision tree is trained and that states that represent stød-bearing phones in many cases resist state-tying or that states are tied around stød. The increased RTF is caused by the increased complexity of the AM, but adding pitch-related features compensate for the increased RTF.

We therefore conclude that stød improves Danish ASR when modelled explicitly in the phonetic dictionary and when modelled acoustically with pitch-related features, leading to significant improvement for both MFCC and PLP-based ASR systems.

### 7.1.4 Stød and voice quality features in Danish speech recognition

We combine the findings in Chapter 4 on the salience of particular acoustic features for stød detection with the findings in Chapter 5 on modelling stød in ASR to improve WER performance. In the experiments, we consider Peak Slope and phase features in addition to a new feature: Harmonic Richness Factor (HRF). In the first study using speaker-adapted GMM-based ASR systems trained on LDA-projected features, the ASR system trained on MFCC features extended with the Peak Slope feature shows degrades performance,

while no significant difference is found for the remaining systems. However, the DNN-based systems based on HRF and phase features achieve the significant improvements WER performance. A surprising increase in decoding speed provide yet more WER improvements because the decoder can traverse a larger part of the lattice.

## 7.2 Danish ASR

A part of the stated objective of this thesis is to provide tools, resources and methodology that can stimulate ASR research and development in Denmark and for Danish language. The sprakbanken setup is based on publicly available tools and resources and the system can be used as a reference for further development and teaching purposes. An open source ASR system with open domain data and published methodology has not previously been available for Danish and the hope is that the availability can improve the state of speech technology research and education.

## 7.3 Future work

The work in this thesis does not provide a specific answer to the ephemeral "... more than creak", but points to several acoustic measures such as HRF and phase features as promising research areas that can provide more insight into the nature of stød and improve our understanding of the phenomenon. Below we enumerate the research we believe to be able to further the characterisation of stød and can improve ASR for Danish spoken language in general:

1. Segment-based normalisation of features in stød detection to remove the impact of the segment on stød prediction
2. The correlation between stød and phase features
3. Speed and quality of phase feature estimation in ASR
4. Modelling stress in ASR dictionaries to determine the impact on stød modelling
5. Modelling pronuciation variation in ASR dictionaries to determine the impact on ASR performance and stød modelling
6. The impact LDA transformation has on Peak Slope
7. Noise robustness of voice quality features and stød modelling

8. Incorporating the remaining data developed by Nordisk Språkteknologi in the sprakbanken recipe
9. Modelling compounding in ASR to reduce the size of the dictionary
10. Multi-lingual or inter-Scandinavian ASR

Because stød can distinguish lexemes and especially word classes, parsing, named entity recognition and disambiguation and other NLP systems used in spoken language understanding can potentially benefit from the added information. It remains to be determined whether this is possible for ASR systems with pronunciation variants.

With respect to the sprakbanken setup, it is important that more resources from the Språkbanken corpus is exploited. This includes the 8 kHz and 22 kHz data in the Danish part and also the Swedish and Norwegian parts of the corpus. A recipe that uses the Swedish part of Språkbanken is currently under development at KTH, Sweden. The impact of using the oft-mentioned phonetic dictionary created together with the Språkbanken corpus needs to be evaluated against the eSpeak-generated dictionary. We also believe that a different stratification of the test data will make sense from an application and academic point of view.

## 7.4 Final conclusions

We have shown that it is possible to use the distinguishing function of stød to improve Danish automatic speech recognition. To reach this conclusion, we have

1. determined that stød annotation is reliable
2. identified 18 acoustic features that carry important information which signal stød
3. confirmed that stød can be detected in acoustic features when stød is detected jointly with the underlying segment
4. demonstrated that GMM-based and DNN-based ASR systems that model stød outperform similar systems that do not, if the ASR systems are trained on LDA-projected MFCC features
5. combined stød-informed ASR with stød-related acoustic features to show additional performance improvement

This is the first data-driven acoustic investigation of stød and the zero-knowledge approach has opened new avenues of research by showing correlation between stød and phase features. A large by-product of the present work is a free and open source automatic speech recogniser for Danish that we hope will be used by others to further teaching and research.

# References


Adda-Decker, M., & Adda, G. (2000). Morphological decomposition for ASR in German. In *Workshop on Phonetics and Phonology in ASR, Saarbrücken, Germany* (pp. 129–143).

Allauzen, C., Mohri, M., Riley, M., & Roark, B. (2004). A generalized construction of integrated speech recognition transducers. In *Acoustics, Speech, and Signal Processing, 2004. Proceedings.(ICASSP'04). IEEE International Conference on* (Vol. 1, pp. I–761).

Allen, J. B., & Rabiner, L. R. (1977). A unified approach to short-time Fourier analysis and synthesis. *Proceedings of the IEEE*, *65*(11), 1558–1564.

Andersen, G. (2008). Akustiske databser for dansk. *(http://www.nb.no/sbfil/dok/nst_taledat_dk.pdf (Norwegian))*.

Barsøe Management. (2008). *Klinisk og ressourcemæssig evaluering af MIRSK digital diktering på Rigshospitalet.* (Avaliable upon request from MIRSK Digital).

Basapur, S., Xu, S., Ahlenius, M., & Lee, Y. S. (2007). User expectations from dictation on mobile devices. In *Human-Computer Interaction. Interaction Platforms and Techniques* (pp. 217–225). Springer.

Boersma, P. (1993). Accurate short-term analysis of the fundamental frequency and the harmonics-to-noise ratio of a sampled sound. In *Proceedings of the institute of phonetic sciences* (Vol. 17, pp. 97–110).

Boersma, P. (2002). Praat, a system for doing phonetics by computer. *Glot international*, *5*(9/10), 341–345.

Bőhm, T. M. (2009). Analysis and modeling of speech produced with irregular phonation. *Budapesti Műszaki és Gazdaságtudományi Egyetem*.

Chan, D., Fourcin, A., Gibbon, D., Granstrom, B., Huckvale, M., Kokkinakis, G., ... Moreno, A. (1995). EUROM-A spoken language resource for the EU. In *Proceedings of the 4th European Conference on Speech Communication and Speech Tecnology, Eurospeech'95* (pp. 867–880).

Chang, Chih-Chung and Lin, Chih-Jen. (2011). LIBSVM: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, *2*(3), 27.

Cohen, J. (1960). A coefficient of agreement for nominal scale. *Educ Psychol Meas*, *20*, 37–46.

Degottex, G., & Erro, D. (2014). A uniform phase representation for the harmonic model in speech synthesis applications. *EURASIP Journal on Audio, Speech, and Music Processing*, *2014*(1), 1–16.

Degottex, G., Kane, J., Drugman, T., Raitio, T., & Scherer, S. (2014). COVAREP—A collaborative voice analysis repository for speech technologies. In *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on* (pp. 960–964).

Degottex, G., & Stylianou, Y. (2013). Analysis and synthesis of speech using an adaptive full-band harmonic model. *Audio, Speech, and Language Processing, IEEE Transactions on*, *21*(10), 2085–2095.

Drugman, T., & Dutoit, T. (2010). Glottal-based analysis of the lombard effect. In *INTERSPEECH* (pp. 2610–2613).

Duddington, J. (2012). *eSpeak Text to speech.* Web publication: `http://espeak.sourceforge.net/`.

Fan, R.-E., Chang, K.-W., Hsieh, C.-J., Wang, X.-R., & Lin, C.-J. (2008). LIBLINEAR: A library for large linear classification. *The Journal of Machine Learning Research*, *9*, 1871–1874.

Fant, G. (1995). The LF-model revisited. Transformations and frequency domain analysis. *Speech Trans. Lab. Q. Rep., Royal Inst. of Tech. Stockholm*, *2*(3), 40.

Federico, M., Bertoldi, N., & Cettolo, M. (2008). IRSTLM: an open source toolkit for handling large scale language models. In *INTERSPEECH* (pp. 1618–1621).

Federico, M., & De Mori, R. (1998). Language modelling. *Spoken Dialogues with Computers*, 199–230.

Feinstein, A. R., & Cicchetti, D. V. (1990). High agreement but low kappa: I. The problems of two paradoxes. *Journal of clinical epidemiology*, *43*(6), 543–549.

Fernandez, R. (2003). *A computational model for the automatic recognition of affect in speech* (Unpublished doctoral dissertation). Unpublished doctoral dissertation, Massachusetts Institute of Technology.

Fernandez, R., Cui, J., Rosenberg, A., Ramabhadran, B., & Cui, X. (2014). Exploiting vocal-source features to improve ASR accuracy for low-resource languages. In *INTERSPEECH* (pp. 805–809).

Fischer-Jørgensen, E. (1989). *A phonetic study of the stød in Standard Danish.* University of Turku, Phonetics.

Fiscus, J. (1998). Sclite Scoring Package Version 1.5. *US National Institute of Standard Technology (NIST), URL http://www. itl. nist. gov/iaui/894.01/tools*.

Fiscus, J. (2007). *Speech recognition scoring toolkit ver. 2.3 (sctk).*

Frazier, M. (2013). The phonetics of Yucatec Maya and the typology of laryngeal complexity. *STUF-Language Typology and Universals Sprachtypologie und Universalienforschung*, *66*(1), 7–21.

Gerhard, D. (2003). *Pitch extraction and fundamental frequency: History and current techniques*. Regina: Department of Computer Science, University of Regina.

Geurts, P., Ernst, D., & Wehenkel, L. (2006). Extremely randomized trees. *Machine learning*, *63*(1), 3–42.

Ghahremani, P., BabaAli, B., Povey, D., Riedhammer, K., Trmal, J., & Khudanpur, S. (2014). A pitch extraction algorithm tuned for automatic speech recognition. In *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on* (pp. 2494–2498).

Gillick, L., & Cox, S. J. (1989). Some statistical issues in the comparison of speech recognition algorithms. In *Acoustics, Speech, and Signal Processing, 1989. ICASSP-89., 1989 International Conference on* (pp. 532–535).

Gjørup, J. (2010). *Lægesekretærer, Effektive Arbejdsprocesser og patientforløb. Inspirationskatalog med Dokumenterede, Erfaringer, Eksempler og Redskaber – Til lægesekretærer og ledere på sygehusafdelinger*. Center for Kvalitetsudvikling. (Web publication: `http://vis.dk/system/files/IdekatalogLaegesekretaererVersion22april2010.pdf`).

Godfrey, J. J., Holliman, E. C., & McDaniel, J. (1992). SWITCHBOARD: Telephone speech corpus for research and development. In *Acoustics, Speech, and Signal Processing, 1992. ICASSP-92., 1992 IEEE International Conference on* (Vol. 1, pp. 517–520).

Gordon, M., & Ladefoged, P. (2001). Phonation types: a cross-linguistic overview. *Journal of Phonetics*, *29*(4), 383–406.

Grønnum, N. (2005). Fonetik og Fonologi, 3. udg. *Akademisk Forlag, København*.

Grønnum, N. (2006). DanPASS-a Danish phonetically annotated spontaneous speech corpus. In *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC), Genova, Italy, May.*

Grønnum, N. (2009). A Danish phonetically annotated spontaneous speech corpus (DanPASS). *Speech Communication*, *51*(7), 594–603.

Grønnum, N., & Basbøll, H. (2001). Consonant length, stød and morae in standard Danish. *Phonetica*, *58*(4), 230–253.

Grønnum, N., & Basbøll, H. (2002). Stød and Length: Acoustic and Cognitive Reality? In *Speech Prosody 2002, International Conference.*

Grønnum, N., & Basbøll, H. (2003). Two-phased stød vowels—a cognitive reality. *Fonetik2003, Lövånger, Phonum*, *9*, 33–6.

Grønnum, N., & Basbøll, H. (2007). Danish Stød–phonological and cognitive issues. *Experimental approaches to phonology*, 192–206.

Grønnum, N., & Basbøll, H. (2012). Danish stød-Towards simpler structural principles? In *Language, Context and Cognition.* Walter de Gruyter.

Grønnum, N., Vazquez-Larruscain, M., & Basbøll, H. (2013). Danish stød: laryngealization or tone. *Phonetica*, *70*(1-2), 66–92.

Gut, U., & Bayerl, P. S. (2004). Measuring the reliability of manual annotations of speech corpora. In *Speech Prosody 2004, International Conference.*

Hansen, G. (2015). *Stød og stemmekvalitet: En akustisk-fonetisk undersøgelse af ændringer i stemmekvaliteten i forbindelse med stød* (Unpublished doctoral dissertation). (Indeholder lydfiler)

Henrichsen, P. (2007). The Danish PAROLE corpus-a merge of speech and writing. *Current Trends in Research on Spoken Language in the Nordic Countries*, *2*, 84–93.

Henrichsen, P. (2014). Phonix: Danish grapheme-to-phoneme transcriber. *To appear in Translation in transition: between cognition, computing and technology*, *1*.

Henrichsen, P., & Kirkedal, A. (2011). Founding a Large-Vocabulary Speech Recognizer for Danish. In *Speech in Action* (pp. 175–193).

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

Hinton, G., Deng, L., Yu, D., Dahl, G., rahman Mohamed, A., Jaitly, N., ... Kingsbury, B. (2012). Deep neural networks for acoustic modeling in speech recognition. *Signal Processing Magazine*.

Hovy, D., Berg-Kirkpatrick, T., Vaswani, A., & Hovy, E. H. (2013). Learning Whom to Trust with MACE. In *HLT-NAACL* (pp. 1120–1130).

Hwang, M.-Y., Lei, X., Ng, T., Bulyko, I., Ostendorf, M., Stolcke, A., ... others (2004). Progress on mandarin conversational telephone speech recognition. In *Chinese Spoken Language Processing, 2004 International Symposium on* (pp. 1–4).

Høysgaard, J. P. (1743). *Concordia res parvæ crescunt, eller Anden Prøve af Dansk Orthographie.* Groth. [Reprinted in Danske Grammatikere, H. Bertelsen (ed), vol. IV, 217–247. Copenhagen: Gyldendal, 1920, and Copenhagen: Det Danske Sprog-og Litteraturselskab, CA Reitzel 1979].

Implement. (2009). *Kortlægning og måling af administrative opgaver: Resultater af målingen på sygehusområdet.* Ministeriet for Sundhed og Forebyggelse. Retrieved from `https://books.google.dk/books?id=XV1CtwAACAAJ`.

International Phonetic Association. (1999). *Handbook of the International Phonetic Association: A guide to the use of the International Phonetic Alphabet*. Cambridge University Press.

Jurafsky, D., & Martin, J. (2008). Speech and language processing (Prentice Hall series in Artificial Intelligence).

Jurgec, P. (2007). Creaky voice in Slovene. In *III Congresso Internacional de Fonética Experimental. Santiago de Compostela: Xunta de Galicia* (pp. 407–420).

Kane, J., & Gobl, C. (2011). Identifying Regions of Non-Modal Phonation Using Features of the Wavelet Transform. In *INTERSPEECH* (pp. 177–180).

Keating, P., Garellek, M., & Kreiman, J. (2014). Acoustic properties of different kinds of creaky voice.

Kirkedal, A. S. (2013). Analysis of phonetic transcriptions for Danish automatic speech recognition. *Proceedings of the 19th Nordic Conference of Computational Linguistics (NODALIDA 2013) May 22–24, 2013, Oslo University, Norway.*, *NEALT Proceedings Series 16*.

Kirkedal, A. S. (2014). Automatic Phonetic Transcription for Danish Speech Recognition. In *2014 CRITT-WCRE Conference.*

Kirshenbaum, E. (2001). Representing IPA phonetics in ASCII. *URL: `http://www.kirshenbaum.net/IPA/ascii-ipa.pdf` (unpublished), Hewlett-Packard Laboratories*.

Li, A., Zheng, F., Byrne, W., Fung, P., Kamm, T., Liu, Y., ... Chen, X. (2000). CASS: a phonetically transcribed corpus of mandarin spontaneous speech. In *INTERSPEECH* (pp. 485–488).

Liu, X., Gales, M. J. F., Hieronymus, J. L., & Woodland, P. C. (2011). Investigation of acoustic units for LVCSR systems. In *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on* (pp. 4872–4875).

Lu, L., Ghoshal, A., & Renals, S. (2013). Acoustic data-driven pronunciation lexicon for large vocabulary speech recognition. In *Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on* (pp. 374–379).

Martinez, M. G., Singla, K., Tammewar, A., Mesa-Lao, B., Thakur, A., Anusuya, M., ... Carl, M. (2014). SEECAT: ASR & Eye-tracking Enabled Computer Assisted Translation. In *The 17th Annual Conference of the European Association for Machine Translation, EAMT 2014* (pp. 81–88).

Matějka, P., Burget, L., Schwarz, P., & Černockỳ, J. (2006). Brno university of technology system for NIST 2005 language recognition evaluation. In *Speaker and Language Recognition Workshop, 2006. IEEE Odyssey 2006: The* (pp. 1–7).

McKinney, W. (2012). *Python for data analysis: Data wrangling with Pandas, NumPy, and IPython.* " O'Reilly Media, Inc.".

Mohri, M., Pereira, F., & Riley, M. (2008). Speech recognition with weighted finite-state transducers. In *Springer Handbook of Speech Processing* (pp. 559–584). Springer.

Novotney, S., & Callison-Burch, C. (2010). Cheap, fast and good enough: Automatic speech recognition with non-expert transcription. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics* (pp. 207–215).

Nowak, S., & Rüger, S. (2010). How reliable are annotations via crowdsourcing: a study about inter-annotator agreement for multi-label image annotation. In *Proceedings of the international conference on Multimedia information retrieval* (pp. 557–566).

Palmer, F. (1968). *Selected papers of JR Firth (1952–1959).* London Longman.

Panayotov, V., Chen, G., Povey, D., & Khudanpur, S. (2015). Librispeech: an ASR corpus based on public domain audio books. In *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on* (pp. 5206–5210).

Passonneau, R. J., & Carpenter, B. (2014). The benefits of a model of annotation. *Transactions of the Association for Computational Linguistics*, *2*, 311–326.

Paul, D. B., & Baker, J. M. (1992). The design for the Wall Street Journal-based CSR corpus. In *Proceedings of the workshop on Speech and Natural Language* (pp. 357–362).

Pausch, R., & Leatherby, J. H. (1991). An empirical study: Adding voice input to a graphical editor. In *Journal of the American Voice Input/Output Society.*

Pedersen, B. S., Wedekind, J., Bøhm-Andersen, S., Henrichsen, P. J., Hoffensetz-Andresen, S., Kirchmeier-Andersen, S., ... Thomsen, H. E. (2012). *Det danske sprog i den digitale tidsalder – The Danish Language in the Digital Age.* Springer. (Available online at `http://www.meta-net.eu/whitepapers`)

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... et al (2011). Scikit-learn: Machine learning in Python. *The Journal of Machine Learning Research*, *12*, 2825–2830.

Placeway, P., Chen, S., Eskenazi, M., Jain, U., Parikh, V., Raj, B., ... Siegler, M. e. a. (1997). The 1996 HUB-4 sphinx-3 system. In *Proc. DARPA Speech recognition workshop* (pp. 85–89).

Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., ... Vesely, K. (2011, December). The Kaldi Speech Recognition Toolkit. In *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding.* IEEE Signal Processing Society. (IEEE Catalog No.: CFP11SRW-USB)

Povey, D., Zhang, X., & Khudanpur, S. (2014). Parallel training of deep neural networks with natural gradient and parameter averaging. *CoRR, vol. abs/1410.7455*.

Prasad, N. V., & Umesh, S. (2013). Improved cepstral mean and variance normalization using Bayesian framework. In *Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on* (pp. 156–161).

Rabiner, L., & Juang, B. (1986). An introduction to hidden Markov models. *ASSP Magazine, IEEE*, *3*(1), 4–16.

Rajnoha, J., & Pollák, P. (2011). ASR systems in noisy environment: Analysis and solutions for increasing noise robustness. *Radioengineering*, *20*(1), 74–83.

Rath, S. P., Povey, D., Veselỳ, K., & Cernockỳ, J. (2013). Improved feature processing for deep neural networks. In *INTERSPEECH* (pp. 109–113).

Redi, L., & Shattuck-Hufnagel, S. (2001). Variation in the realization of glottalization in normal speakers. *Journal of Phonetics*, *29*(4), 407–429.

Riber Petersen, P. (1973). An instrumental investigation of the Danish stød. *Annual Report of the Institute of Phonetics, University of Copenhagen*, *7*, 195–234.

Riedhammer, K., Van Hai Do, J. H., & Hieronymus, J. (2013). A study on LVCSR and keyword search for Tagalog. In *INTERSPEECH* (pp. 2529–2533).

Rybach, D., Hahn, S., Lehnen, P., Nolden, D., Sundermeyer, M., Tüske, Z., ... Ney, H. (2011). RASR-The RWTH Aachen University open source speech recognition toolkit. In *Proc. IEEE Automatic Speech Recognition and Understanding Workshop*.

Salimbajevs, A., & Strigins, J. (2015). Using sub-word n-gram models for dealing with OOV in large vocabulary speech recognition for Latvian. In *Nordic Conference of Computational Linguistics NODALIDA 2015* (p. 281).

Schroeder, T., Schulze, S., Hilsted, J., & Aldershvile, J. (2003). *Basisbog i medicin og kirurgi*. Munksgaard.

Stolcke, A. (2002). SRILM-an extensible language modeling toolkit. In *Proceedings of the international conference on spoken language processing* (Vol. 2, pp. 901–904).

Stolcke, A., Coccaro, N., Bates, R., Taylor, P., Van Ess-Dykema, C., Ries, K., ... Meteer, M. (2000). Dialogue act modeling for automatic tagging and recognition of conversational speech. *Computational linguistics*, *26*(3), 339–373.

Talkin, D. (1995). A robust algorithm for pitch tracking (RAPT). *Speech coding and synthesis*, *495*, 518.

Togneri, R., Toh, A. M., & Nordholm, S. (2006). Evaluation and modification of cepstral moment normalization for speech recognition in additibe Babble ensemble. In *Proc. SST* (pp. 94–99).

Vanhainen, N., & Salvi, G. (2014). Free Acoustic and Language Models for Large Vocabulary Continuous Speech Recognition in Swedish. *training*, *965*(307568), 420–8.

Wells, J. e. a. (1997). SAMPA computer readable phonetic alphabet. *Handbook of standards and resources for spoken language systems*, *4*.

Witten, I. H., & Bell, T. C. (1991). The zero-frequency problem: Estimating the probabilities of novel events in adaptive text compression. *Information Theory, IEEE Transactions on*, *37*(4), 1085–1094.

Woodland, P. C. (2001). Speaker adaptation for continuous density HMMs: A review. In *ISCA Tutorial and Research Workshop (ITRW) on Adaptation Methods for Speech Recognition.*

Yoon, T.-J., Zhuang, X., Cole, J., & Hasegawa-Johnson, M. (2006). Voice quality dependent speech recognition. In *International Symposium on Linguistic Patterns in Spontaneous Speech.*

Young, S. J. (1993). *The HTK Hidden Markov model toolkit: Design and philosophy.* University of Cambridge, Department of Engineering.

Young, S. J., Odell, J. J., & Woodland, P. C. (1994). Tree-based state tying for high accuracy acoustic modelling. In *Proceedings of the workshop on Human Language Technology* (pp. 307–312).

Zapata, J., & Kirkedal, A. S. (2015). Assessing the Performance of Automatic Speech Recognition Systems When Used by Native and Non-Native Speakers of Three Major Languages in Dictation Workflows. In *Nordic Conference of Computational Linguistics NODALIDA 2015* (p. 201).

# Appendix A

# Appendix

| Segment | Train set | Test set | Segment | Train set | Test set |
|---|---|---|---|---|---|
| eː | 2049 | 14 | ðˠ | 0 | 14 |
| eːˀ | 2780 | 14 | ðˠˀ | 0 | 14 |
| l | 31862 | 159 | ŋ | 3623 | 20 |
| lˀ | 2341 | 26 | ŋˀ | 2158 | 7 |
| m | 23470 | 232 | œː | 0 | 4 |
| mˀ | 2463 | 5 | œːˀ | 0 | 6 |
| n | 50547 | 155 | ɐ̯ | 7649 | 49 |
| nˀ | 10194 | 58 | ɐ̯ˀ | 2557 | 51 |
| oː | 1894 | 28 | ɑː | 3067 | 83 |
| oːˀ | 1292 | 23 | ɑːˀ | 653 | 40 |
| yː | 608 | 6 | ɒ | 12707 | 6 |
| yːˀ | 425 | 8 | ɒˀ | 1409 | 5 |
| æː | 5409 | 7 | ɒː | 2594 | 14 |
| æːˀ | 0 | 44 | ɒːˀ | 15 | 30 |
| ðˠ | 0 | 50 | ɔː | 1545 | 31 |
| ðˠˀ | 0 | 15 | ɔːˀ | 1030 | 25 |

Table A.1: Minimal pairs with respect to stød distribution.

| Phones | | | Features sets | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | Full | | PLP | | Select+ | |
| Classes | | Samples | Accuracy | +/- | Accuracy | +/- | Accuracy | +/- |
| aˀ | a | 119 | 0.736 | 0.09 | 0.790 | 0.10 | 0.737 | 0.23 |
| aːˀ | aː | 11 | 0.850 | 0.40 | 0.950 | 0.20 | 1.000 | 0.00 |
| cːˀ | cː | 602 | 0.784 | 0.15 | 0.788 | 0.07 | 0.814 | 0.10 |
| eˀ | e | 3977 | 0.727 | 0.06 | 0.752 | 0.04 | 0.734 | 0.05 |
| eːˀ | eː | 2780 | 0.865 | 0.02 | 0.788 | 0.03 | 0.881 | 0.04 |
| iˀ | i | 2281 | 0.767 | 0.04 | 0.786 | 0.05 | 0.756 | 0.06 |
| iːˀ | iː | 1292 | 0.855 | 0.05 | 0.807 | 0.02 | 0.867 | 0.04 |
| jˀ | j | 1539 | 0.876 | 0.03 | 0.888 | 0.03 | 0.867 | 0.04 |
| lˀ | l | 2341 | 0.722 | 0.21 | 0.749 | 0.12 | 0.720 | 0.18 |
| mˀ | m | 2463 | 0.743 | 0.06 | 0.739 | 0.05 | 0.735 | 0.07 |
| m̩ˀ | m̩ | 9 | 1.000 | 0.00 | 0.750 | 0.32 | 1.000 | 0.00 |
| nˀ | n | 10194 | 0.754 | 0.22 | 0.723 | 0.19 | 0.742 | 0.23 |
| oˀ | o | 1931 | 0.837 | 0.04 | 0.854 | 0.07 | 0.862 | 0.08 |
| oːˀ | oː | 1292 | 0.890 | 0.05 | 0.815 | 0.08 | 0.897 | 0.07 |
| qːˀ | qː | 478 | 0.805 | 0.02 | 0.692 | 0.05 | 0.794 | 0.10 |
| uˀ | u | 3133 | 0.763 | 0.07 | 0.745 | 0.08 | 0.768 | 0.04 |
| uːˀ | uː | 354 | 0.788 | 0.14 | 0.829 | 0.03 | 0.818 | 0.10 |
| wˀ | w | 2332 | 0.678 | 0.11 | 0.661 | 0.09 | 0.674 | 0.12 |
| xːˀ | xː | 21 | 0.850 | 0.19 | 0.800 | 0.41 | 0.815 | 0.27 |
| yˀ | y | 92 | 0.738 | 0.12 | 0.733 | 0.21 | 0.739 | 0.18 |
| yːˀ | yː | 425 | 0.708 | 0.16 | 0.678 | 0.09 | 0.758 | 0.21 |
| zːˀ | zː | 2168 | 0.770 | 0.06 | 0.703 | 0.04 | 0.810 | 0.03 |
| æˀ | æ | 3268 | 0.728 | 0.05 | 0.728 | 0.06 | 0.719 | 0.07 |
| ðˀ | ð | 5336 | 0.726 | 0.26 | 0.664 | 0.23 | 0.711 | 0.27 |
| øˀ | ø | 190 | 0.771 | 0.10 | 0.803 | 0.05 | 0.789 | 0.15 |
| ŋˀ | ŋ | 2158 | 0.744 | 0.14 | 0.702 | 0.16 | 0.729 | 0.16 |
| œˀ | œ | 177 | 0.876 | 0.08 | 0.896 | 0.10 | 0.907 | 0.01 |
| ɐˀ | ɐ | 11 | 0.600 | 0.51 | 0.700 | 0.37 | 0.917 | 0.21 |
| ɐ̯ˀ | ɐ̯ | 2557 | 0.740 | 0.07 | 0.706 | 0.07 | 0.744 | 0.08 |
| ɑˀ | ɑ | 2925 | 0.756 | 0.06 | 0.754 | 0.04 | 0.743 | 0.07 |
| ɑːˀ | ɑː | 653 | 0.860 | 0.04 | 0.822 | 0.07 | 0.839 | 0.08 |
| ɒˀ | ɒ | 1409 | 0.740 | 0.06 | 0.770 | 0.09 | 0.734 | 0.09 |
| ɒːˀ | ɒː | 15 | 0.733 | 0.54 | 0.767 | 0.27 | 0.867 | 0.39 |
| ɔˀ | ɔ | 1912 | 0.791 | 0.06 | 0.798 | 0.06 | 0.778 | 0.10 |
| ɔːˀ | ɔː | 1030 | 0.754 | 0.10 | 0.706 | 0.12 | 0.765 | 0.11 |
| ɛˀ | ɛ | 177 | 0.647 | 0.23 | 0.653 | 0.22 | 0.718 | 0.15 |
| ɛːˀ | ɛː | 296 | 0.750 | 0.12 | 0.725 | 0.16 | 0.784 | 0.24 |
| ɲˀ | ɲ | 392 | 0.702 | 0.10 | 0.728 | 0.15 | 0.684 | 0.15 |
| œˀ | œ | 314 | 0.616 | 0.04 | 0.758 | 0.09 | 0.664 | 0.14 |
| ʁˀ | ʁ | 1575 | 0.925 | 0.02 | 0.899 | 0.03 | 0.922 | 0.02 |
| ʌˀ | ʌ | 13 | 0.867 | 0.33 | 0.767 | 0.12 | 0.917 | 0.21 |
| Mean classification accuracy | | | 0.781 | 0.168 | 0.769 | 0.144 | **0.803** | **0.176** |

Table A.2: 5-fold One-vs-One evaluation on training data using different feature sets.

| Phones | | | Features sets | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | Full | | PLP | | Select+ | |
| Classes | | Samples | Accuracy | +/- | Accuracy | +/- | Accuracy | +/- |
| a$^{?}$ | a | 130 | 0.946 | 0.05 | 0.908 | 0.05 | 0.873 | 0.05 |
| c$^{?}$ | c | 602 | 0.906 | 0.02 | 0.874 | 0.05 | 0.836 | 0.02 |
| e$^{?}$ | e | 6757 | 0.868 | 0.02 | 0.814 | 0.01 | 0.766 | 0.01 |
| i$^{?}$ | i | 3573 | 0.885 | 0.03 | 0.841 | 0.02 | 0.788 | 0.03 |
| j$^{?}$ | j | 1539 | 0.972 | 0.02 | 0.955 | 0.03 | 0.933 | 0.02 |
| l$^{?}$ | l | 2341 | 0.910 | 0.02 | 0.877 | 0.02 | 0.837 | 0.01 |
| m$^{?}$ | m | 2472 | 0.923 | 0.01 | 0.891 | 0.02 | 0.855 | 0.01 |
| n$^{?}$ | n | 10194 | 0.920 | 0.01 | 0.885 | 0.01 | 0.844 | 0.01 |
| o$^{?}$ | o | 3223 | 0.939 | 0.01 | 0.900 | 0.01 | 0.843 | 0.02 |
| q$^{?}$ | q | 478 | 0.903 | 0.02 | 0.801 | 0.07 | 0.807 | 0.05 |
| u$^{?}$ | u | 3487 | 0.906 | 0.01 | 0.867 | 0.02 | 0.806 | 0.03 |
| w$^{?}$ | w | 2332 | 0.938 | 0.02 | 0.873 | 0.02 | 0.838 | 0.02 |
| x$^{?}$ | x | 21 | 0.950 | 0.12 | 0.975 | 0.10 | 0.950 | 0.12 |
| y$^{?}$ | y | 517 | 0.896 | 0.04 | 0.831 | 0.08 | 0.830 | 0.04 |
| z$^{?}$ | z | 2168 | 0.930 | 0.02 | 0.868 | 0.02 | 0.824 | 0.02 |
| æ$^{?}$ | æ | 3268 | 0.915 | 0.01 | 0.867 | 0.03 | 0.802 | 0.03 |
| ð$^{?}$ | ð | 5336 | 0.917 | 0.01 | 0.869 | 0.01 | 0.797 | 0.02 |
| ø$^{?}$ | ø | 190 | 0.953 | 0.09 | 0.900 | 0.13 | 0.918 | 0.09 |
| ŋ$^{?}$ | ŋ | 2158 | 0.950 | 0.01 | 0.919 | 0.01 | 0.883 | 0.01 |
| œ$^{?}$ | œ | 177 | 0.980 | 0.02 | 0.957 | 0.04 | 0.929 | 0.05 |
| ɐ$^{?}$ | ɐ | 2568 | 0.897 | 0.01 | 0.846 | 0.01 | 0.774 | 0.02 |
| ɑ$^{?}$ | ɑ | 3578 | 0.901 | 0.01 | 0.855 | 0.02 | 0.818 | 0.02 |
| ɒ$^{?}$ | ɒ | 1424 | 0.889 | 0.03 | 0.839 | 0.03 | 0.808 | 0.03 |
| ɔ$^{?}$ | ɔ | 2942 | 0.911 | 0.01 | 0.863 | 0.03 | 0.837 | 0.02 |
| ɛ$^{?}$ | ɛ | 473 | 0.921 | 0.06 | 0.857 | 0.06 | 0.831 | 0.07 |
| ɲ$^{?}$ | ɲ | 392 | 0.944 | 0.04 | 0.852 | 0.06 | 0.835 | 0.03 |
| ɶ$^{?}$ | ɶ | 314 | 0.865 | 0.07 | 0.857 | 0.09 | 0.809 | 0.05 |
| ʁ$^{?}$ | ʁ | 1575 | 0.956 | 0.02 | 0.939 | 0.02 | 0.942 | 0.02 |
| ʌ$^{?}$ | ʌ | 13 | 0.967 | 0.13 | 0.967 | 0.13 | 0.967 | 0.13 |
| Mean classification accuracy | | | 0.922 | 0.058 | 0.885 | 0.096 | 0.853 | 0.119 |

Table A.3: 5-fold One-vs-One evaluation on training data using different feature sets and coarser annotation.

# Appendix B

# ASR resources

## B.1 Software and scripts

The scripts used to extract features and annotation using Praat is available on GitHub at `https://github.com/dresen/praat`. The software used for the classification experiments can be installed by following the instructions here: `http://scikit-learn.org/stable/install.html`

## B.2 Kaldi

The Kaldi ASR setup can be downloaded from GitHub at `https://github.com/kaldi-asr/kaldi` and the setup itself can be viewed here `https://github.com/kaldi-asr/kaldi/tree/master/egs/sprakbanken`. The installation of the Kaldi toolkit can be difficult and I recommend reserving some time for it, especially if you do not have super user privileges while installing. For Kaldi to work, CPU throttling must be disabled.

The training recipe is included here for reference and because the recipe is under continuous development by the Kaldi community. The current sprakbanken recipe diverges somewhat from the recipe in Section B.2.1 because it does not use voice quality feature augmentation. Be aware that the parallellisation settings are hard-coded for the server used in the experiments. To choose feature type, pass it as an argument on the command line, e.g. `./run.sh plp_pitch`.

### B.2.1 ASR training script

Listing B.1: ASR training script

```
#!/bin/bash

. ./cmd.sh ## You'll want to change cmd.sh to something that will work on your system.
           ## This relates to the queue.
. ./path.sh # so python3 is on the path if not on the system (we made a link to utils/).a

# This is a shell script, but it's recommended that you run the commands one by
# one by copying and pasting into the shell.


# Download the corpus and prepare parallel lists of sound files and text files
# Divide the corpus into train, dev and test sets
local/sprak_data_prep.sh  || exit 1;

# Perform text normalisation, prepare dict folder and LM data transcriptions
# This setup uses previsously prepared data. eSpeak must be installed and in PATH to use dict_prep.sh
#local/dict_prep.sh || exit 1;
local/copy_dict.sh || exit 1;


utils/prepare_lang.sh data/local/dict "<UNK>" data/local/lang_tmp data/lang || exit 1;

# Declares the type of feature vectors (MFCC or PLP +/- pitch)
subexp=$1


# Extract feature vectors
# p was added to the rspecifier (scp,p:$logdir/wav.JOB.scp) in make_mfcc.sh because some
# wave files are corrupt
# Will return a warning message because of the corrupt audio files, but compute them anyway
# If this step fails and prints a partial diff, rerun from sprak_data_prep.sh

steps/make_$subexp.sh --nj 10 --cmd $train_cmd data/$subexp/test exp/$subexp/make_$subexp/test $subexp &
steps/make_$subexp.sh --nj 10 --cmd $train_cmd data/$subexp/dev exp/$subexp/make_$subexp/dev $subexp &
steps/make_$subexp.sh --nj 10 --cmd $train_cmd data/$subexp/train exp/$subexp/make_$subexp/train $subexp
     || exit 1;
wait

# Compute cepstral mean and variance normalisation
steps/compute_cmvn_stats.sh data/$subexp/test exp/$subexp/make_$subexp/test $subexp &
steps/compute_cmvn_stats.sh data/$subexp/dev exp/$subexp/make_$subexp/dev $subexp &
steps/compute_cmvn_stats.sh data/$subexp/train exp/$subexp/make_$subexp/train $subexp

wait

# Repair data set (remove corrupt data points with corrupt audio)

utils/fix_data_dir.sh data/$subexp/test &
utils/fix_data_dir.sh data/$subexp/dev &
utils/fix_data_dir.sh data/$subexp/train
wait

```

```
# Train LM with CMUCLMTK
# This setup uses IRSTLM
#local/sprak_train_lm.sh &> data/$subexp/local/cmuclmtk/lm.log

# Train LM with irstlm
local/train_irstlm.sh data/local/transcript_lm/transcripts.uniq 3 "${subexp}3g" data/lang data/local/${
    subexp}3_lm &> data/local/3g.log &

# Now make subset of the training data with the shortest 120k utterances.
utils/subset_data_dir.sh --shortest data/$subexp/train 120000 data/$subexp/train_120kshort || exit 1;

# Train monophone model on short utterances
steps/train_mono.sh --nj 30 --cmd "$train_cmd" \
  data/$subexp/train_120kshort data/lang exp/$subexp/mono0a || exit 1;

# Ensure that LMs are created
wait

utils/mkgraph.sh --mono data/lang_test_${subexp}3g exp/$subexp/mono0a exp/$subexp/mono0a/graph_3g &

# Ensure that all graphs are constructed
wait

steps/decode.sh --nj 7 --cmd "$decode_cmd" \
      exp/$subexp/mono0a/graph_3g data/$subexp/test exp/$subexp/mono0a/decode_3g_test


steps/align_si.sh --nj 30 --cmd "$train_cmd" \
   data/$subexp/train data/lang exp/$subexp/mono0a exp/$subexp/mono0a_ali || exit 1;

steps/train_deltas.sh --cmd "$train_cmd" \
    2000 10000 data/$subexp/train data/lang exp/$subexp/mono0a_ali exp/$subexp/tri1 || exit 1;

wait


utils/mkgraph.sh data/lang_test_${subexp}3g exp/$subexp/tri1 exp/$subexp/tri1/graph_3g &

steps/decode.sh --nj 7 --cmd "$decode_cmd" \
  exp/$subexp/tri1/graph_3g data/$subexp/test exp/$subexp/tri1/decode_3g_test || exit 1;

wait

steps/align_si.sh --nj 30 --cmd "$train_cmd" \
  data/$subexp/train data/lang exp/$subexp/tri1 exp/$subexp/tri1_ali || exit 1;


# Train tri2a, which is deltas + delta-deltas.
steps/train_deltas.sh --cmd "$train_cmd" \
  2500 15000 data/$subexp/train data/lang exp/$subexp/tri1_ali exp/$subexp/tri2a || exit 1;

utils/mkgraph.sh data/lang_test_${subexp}3g exp/$subexp/tri2a exp/$subexp/tri2a/graph_3g || exit 1;

steps/decode.sh --nj 7 --cmd "$decode_cmd" \
  exp/$subexp/tri2a/graph_3g data/$subexp/test exp/$subexp/tri2a/decode_3g_test || exit 1;

```

```
107
108 steps/train_lda_mllt.sh --cmd "$train_cmd" \
109    --splice-opts "--left-context=5␣--right-context=5" \
110    2500 15000 data/$subexp/train data/lang exp/$subexp/tri1_ali exp/$subexp/tri2b || exit 1;
111
112 utils/mkgraph.sh data/lang_test_${subexp}3g exp/$subexp/tri2b exp/$subexp/tri2b/graph_3g || exit 1;
113 steps/decode.sh --nj 7 --cmd "$decode_cmd" \
114   exp/$subexp/tri2b/graph_3g data/$subexp/test exp/$subexp/tri2b/decode_3g_test || exit 1;
115
116
117 steps/align_si.sh  --nj 30 --cmd "$train_cmd" \
118   --use-graphs true data/$subexp/train data/lang exp/$subexp/tri2b exp/$subexp/tri2b_ali  || exit 1;
119
120
121 # From 2b system, train 3b which is LDA + MLLT + SAT.
122 steps/train_sat.sh --cmd "$train_cmd" \
123   2500 15000 data/$subexp/train data/lang exp/$subexp/tri2b_ali exp/$subexp/tri3b || exit 1;
124 utils/mkgraph.sh data/lang_test_${subexp}3g exp/$subexp/tri3b exp/$subexp/tri3b/graph_3g || exit 1;
125 steps/decode_fmllr.sh --nj 7 --cmd "$decode_cmd" \
126   exp/$subexp/tri3b/graph_3g data/$subexp/test exp/$subexp/tri3b/decode_3g_test || exit 1;
127
128
129 # From 3b system
130 steps/align_fmllr.sh --nj 30 --cmd "$train_cmd" \
131   data/$subexp/train data/lang exp/$subexp/tri3b exp/$subexp/tri3b_ali || exit 1;
132
133 # From 3b system, train another SAT system (tri4a) with all the si284 data.
134
135 steps/train_sat.sh  --cmd "$train_cmd" \
136   4200 40000 data/$subexp/train data/lang exp/$subexp/tri3b_ali exp/$subexp/tri4a || exit 1;
137
138 utils/mkgraph.sh data/lang_test_${subexp}3g exp/$subexp/tri4a exp/$subexp/tri4a/graph_3g || exit 1;
139 steps/decode_fmllr.sh --nj 7 --cmd "$decode_cmd" \
140    exp/$subexp/tri4a/graph_3g data/$subexp/test exp/$subexp/tri4a/decode_3g_test || exit 1;
141
142
143 steps/train_sat.sh --cmd "$train_cmd" \
144    4800 60000 data/$subexp/train data/lang exp/$subexp/tri3b_ali exp/$subexp/tri4b || exit 1;
145
146  utils/mkgraph.sh data/lang_test_${subexp}3g exp/$subexp/tri4b exp/$subexp/tri4b/graph_3g || exit 1;
147  steps/decode_fmllr.sh --nj 7 --cmd "$decode_cmd" \
148    exp/$subexp/tri4b/graph_3g data/$subexp/test exp/$subexp/tri4b/decode_3g_test || exit 1;
149
150
151 # alignment used to train nnets and sgmms
152 steps/align_fmllr.sh --nj 30 --cmd "$train_cmd" \
153   data/$subexp/train data/lang exp/$subexp/tri4b exp/$subexp/tri4b_ali || exit 1;
154
155
156 ## Train here, instead of separate script
157 steps/nnet2/train_tanh_fast.sh --initial-learning-rate 0.01 --final-learning-rate 0.001 --num-hidden-
       layers 5 --hidden-layer-dim 1024 --num_jobs_nnet 16 --cmd "$train_cmd" data/$subexp/train data/lang
       exp/$subexp/tri4b_ali/ ${subexp}_nnet5c || exit 1;
158
159 ## Decode here
```

```
steps/nnet2/decode.sh --cmd utils/run.pl  --nj 7 --transform-dir exp/$subexp/tri4b/decode_3g_test exp/
    $subexp/tri4b/graph_3g/ data/$subexp/test/ ${subexp}_nnet5c/decode_3g_test || exit 1;


# Getting results [see RESULTS file]
for x in exp/$subexp/*/decode*; do [ -d $x ] && grep WER $x/wer_* | utils/best_wer.sh; done >
    RESULTS_$subexp
```

## B.3 Covarep feature extraction script

Covarep is available at `https://github.com/covarep/covarep`. The feature extraction used in the experiments diverge slightly in the settings and the output among other things. The lower threshold on FO range was set to 20 Hz to account for non-modal speech which is expected during the pronunciation of stød-bearing phones.

Listing B.2: Covarep feature extraction script

```
% General COVAREP feature extraction script
%
% Description
%  This script extracts features to do with glottal source and spectral
%  envelope available with the COVAREP repository. For each .wav file in
%  the inputted directory path, .mat files are produced containing
%
% Input
%  in_dir [directory path] : Path to directory containing wav files to be analysed
%  sample_rate [seconds] : feature sampling rate in seconds (optional)
%
% Output
%         : No arguments are outputted with this script, though .mat files
%         are saved corresponding to each .wav file. .mat files contain the
%         feature matrix: features [number of frames X 35] and names:
%         containing the feature name correspond to each column of the
%         feature matrix
% Example
%   in_dir='//home/john/Desktop/test_dir'; % Specify directory of wavs
%   sample_rate=0.01; % State feature sampling rate
%   COVAREP_feature_extraction(in_dir,sample_rate); % Launch feature extraction
%
% Copyright (c) 2013 Trinity College Dublin - Phonetics & Speech Lab
%
% License
%  This file is under the LGPL license,  you can
%  redistribute it and/or modify it under the terms of the GNU Lesser General
%  Public License as published by the Free Software Foundation, either version 3
%  of the License, or (at your option) any later version. This file is
%  distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY;
%  without even the implied warranty of MERCHANTABILITY or FITNESS FOR A
%  PARTICULAR PURPOSE. See the GNU Lesser General Public License for more
%  details.
%
% Note
%  This function has been developed and tested on speech signals sampled
%  at 16 kHz. Though the analysis should be sampling frequency independent
%  we cannot guarantee optimal performance on non-16 kHz signals
%
% This function is part of the Covarep project: http://covarep.github.io/covarep
%
% Author
%  John Kane <kanejo@.tcd.ie>


```

```
function ASR_feature_extraction(in_dir,sample_rate,names)

% F0 settings
F0min = 20; % Minimum F0 set to 50 Hz
F0max = 400; % Maximum F0 set to 500 Hz

% IAIF settings
hpfilt = 1;
d = 0.99;

% LP settings
LP_winLen=0.025;
LP_winShift=0.005;

% Rd MSP settings
opt = sin_analysis();
opt.fharmonic  = true;
opt.use_ls     = false;
opt.debug = 0;

% Envelope settings
opt.use_ls     = false; % Use Peak Picking
opt.dftlen     = 4096;  % Force the DFT length
opt.frames_keepspec = true; % Keep the computed spectra in the frames structure

% Analysis settings
fileList=dir([in_dir filesep '*.wav']);
N=length(fileList);

if N==0
    disp('No wav files in inputted directory!!!')
end

%% Do processing
for n=1:N

    basename=regexp(fileList(n).name,'\.wav','split');
    basename=char(basename(1));
    str=sprintf('Analysing file: %s',basename);
    disp(str)
    try
        % Load file and set sample locations
        [x,fs]=wavread([in_dir filesep basename '.wav']);
        feature_sampling=round((sample_rate/2)*fs):round(sample_rate*fs):length(x);

        % Check if signal is mono or stereo
        if(size(x, 2) ~= 1)
            warning(sprintf('file: %s is not a mono signal. processing only first channel.', basename));
            x = x(:,1);
        end

        % Polarity detection
        polarity = polarity_reskew(x,fs);
        x=polarity*x; % Correct polarity if necessary

        % F0/GCI detection
```

```
        [srh_f0,srh_vuv,~,srh_time] = pitch_srh(x,fs,F0min,F0max, ...
            sample_rate*1000);
        F0med=median(srh_f0(srh_f0>F0min&srh_f0<F0max&srh_vuv==1));
        VUV_int = interp1(round(srh_time*fs),srh_vuv,1:length(x));
        VUV_int(isnan(VUV_int)==1)=0;

        GCI = gci_sedreams(x,fs,F0med,1); % SEDREAMS GCI detection
        GCI=round(GCI*fs); GCI(GCI<1|isnan(GCI)==1|isinf(GCI)==1)=[];
        GCI(VUV_int(GCI)<.5)=[]; % Remove GCIs in detected unvoiced regions
        GCI=unique(GCI); % Remove possible duplications

        % Iterative and adaptive inverse filtering (IAIF) & LP inverse
        % filtering
        p_gl = 2*round(fs/4000);
        p_vt = 2*round(fs/2000)+4;
        [g_iaif,gd_iaif] = iaif_gci(x,fs,GCI/fs,p_vt,p_gl,d,hpfilt);

        % Glottal source parameterisation
        [NAQ,QOQ,H1H2,HRF,PSP] = get_vq_params(g_iaif,gd_iaif,fs,GCI/fs); % Estimate conventional
            glottal parameters

        % Wavelet-based parameters
        PS = peakslope(x,fs);   % peakSlope extraction
        PS=interp1(PS(:,1)*fs,PS(:,2),feature_sampling);

        % Rd parameter estimation of the LF glottal model using Mean Squared
        % Phase (MSP)
        srh_f0(srh_f0==0) = 100;
        frames = sin_analysis(x, fs, [srh_time(:),srh_f0(:)], opt);

        % Interpolate features to feature sampling rate
        NAQ=interp1(NAQ(:,1)*fs,NAQ(:,2),feature_sampling);
        QOQ=interp1(QOQ(:,1)*fs,QOQ(:,2),feature_sampling);
        H1H2=interp1(H1H2(:,1)*fs,H1H2(:,2),feature_sampling);
        PSP=interp1(PSP(:,1)*fs,PSP(:,2),feature_sampling);
        HRF=interp1(HRF(:,1)*fs,HRF(:,2),feature_sampling);

        % Add PDM and PDD
        hmpdopt = hmpd_analysis();
        hmpdopt.debug = 0;
        hmpdopt.usemex = false;
        hmpdopt.amp_enc_method=2; hmpdopt.amp_log=true; hmpdopt.amp_order=39;
        hmpdopt.pdd_log=true; hmpdopt.pdd_order=12;% MFCC-like phase variance
        hmpdopt.pdm_log=true; hmpdopt.pdm_order=24;% Number of log-Harmonic coefs
        [hmpdf0s, dummy, HMPDM, HMPDD] = hmpd_analysis_features(frames, fs, hmpdopt);
        HMPDM = irregsampling2uniformsampling(hmpdf0s(:,1), HMPDM, (feature_sampling-1)/fs, @unwrap,
            @wrap, 'linear', 0, hmpdopt.usemex);
        HMPDD = irregsampling2uniformsampling(hmpdf0s(:,1), HMPDD, (feature_sampling-1)/fs, [], [], '
            linear', 0, hmpdopt.usemex);

        % Create feature matrix and save
        features=[NAQ(:) QOQ(:) H1H2(:) PSP(:) HRF(:) PS(:) HMPDM HMPDD];
        features(isnan(features))=0;
        save([in_dir filesep basename '.mat'],'features','names')
        clear features
```

```
155         str=sprintf('.............DONE!!!');
156         disp(str)
157     catch
158         str=sprintf('.............ERROR NOT ANALYSED!!!');
159         disp(str)
160     end
161
162 end
```

## B.4 Stød equivalence classes

This section contains the independent and mixed equivalence classes from tri4b systems that explictly model stød in the phonetic dictionary. The analysis is output by a script written in python which means the phones in an equivalence class are enclosed in quotation marks. If a phone is marked for stress ([’]), the phone is enclosed in double quotation marks (["]), otherwise the phone is enclosed in [’].

### B.4.1 MFCC+stød

#### B.4.1.1 tri4b

```
True [’?O_S’, "’?O_S"]
False [’?u_I’, ’u_I’, "’u_I", "’?u_I"]
False [’?O_I’, ’O_I’, ’O_S’, ’?O_S’, "’O_E", ’?O_B’, "’O_B", "’O_I", ’O_B’,
"’?O_E", "’?O_B", "’?O_S", ’O_E’, "’O_S", ’?O_E’, "’?O_I"]
True ["’?V_I", ’?V_I’]
False [’s_S’, ’?s_S’, ’z_S’]
True ["?’i_S", ’?i_S’]
False [’m_S’, ’?m_S’, ’m_B’, ’?m_B’]
False [’z_E’, ’s_E’, ’?s_E’]
True ["?’i_I", ’?i_I’]
False [’m_I’, ’?m_I’]
False [’?@-_E’, ’@-_E’, "’@-_E"]
True ["’?o_B", ’?o_B’]
True [’?O_I’, "’?O_I"]
True [’?A_B’, "’?A_B"]
True ["?’E_E", "’?E_E", ’?E_E’]
True ["’?e_I", ’?e_I’]
False [’s_I’, ’z_I’, ’?s_I’]
True ["’?V_S", ’?V_S’]
True ["’?&+_S", ’?&+_S’]
True [’?&+_I’, "’?&+_I"]
True ["’?W_B", ’?W_B’]
True ["?’i_E", ’?i_E’]
True ["’?W+_B", ’?W+_B’]
True ["?’E_I", "’?E_I", ’?E_I’]
True ["’?W_I", ’?W_I’]
True ["’?E_B", "?’E_B", ’?E_B’]
True [’?A_E’, "’?A_E"]
True [’?&_E’, "’?&_E"]
True [’?o_S’, "’?o_S"]
True ["’?&_B", ’?&_B’]
True [’?W_E’, "’?W_S", "’?W_E", ’?W_S’]
True [’?W+_S’, ’?W+_E’, "’?W+_S", "’?W+_E"]
True [’?V_B’, "’?V_B"]
True [’?&_S’, "’?&_S"]
```

```
False ['?d_B', 'd_B']
False ['z_B', '?s_B', 's_B']
True ["'?e_B", '?e_B']
True ["'?e_E", '?e_E']
True ['?V_E', "'?V_E"]
True ['?A_I', "'?A_I", '?A_S', "'?A_S"]
True ['?O_E', "'?O_E"]
True ['?W+_I', "'?W+_I"]
False ["'?u_B", '?u_B', 'u_B', "'u_B"]
False ['u_E', '?u_E', "'?u_E", "'u_E"]
True ["'?&_I", '?&_I']
False ['?@-_I', "'@-_B", '?@-_B', '?@-_S', "'@-_S", '@-_I', '@-_S', '@-_B', "'@-_I"]
False ['?d_S', 'd_E', 'd_I', '?d_I', '?d_E', 'd_S']
True ["'?E_S", '?E_S', "?'E_S"]
True ['?e_S', "'?e_S"]
True ["'?y_I", '?y_I']
True ['?O_B', "'?O_B"]
True ["'?&+_E", '?&+_E']
True ['?i_B', "?'i_B"]
True ['?o_E', "'?o_E"]
True ["'?o_I", '?o_I']
False ['?u_S', "'?u_S", 'u_S', "'u_S"]
True ["'?y_E", "'?y_B", '?y_E', "'?y_S", '?y_S', '?y_B']
True ['?&+_B', "'?&+_B"]
```

#### B.4.1.2 nnet5c

```
False ['?m_E', 'm_E']
True ['?O_S', "'?O_S"]
False ['?u_I', 'u_I', "'u_I", "'?u_I"]
False ['?O_I', 'O_I', 'O_S', '?O_S', "'O_E', '?O_B', "'O_B", "'O_I', 'O_B',
"'?O_E", "'?O_B", "'?O_S", 'O_E', "'O_S", '?O_E', "'?O_I"]
True ["'?V_I", '?V_I']
False ['s_S', '?s_S', 'z_S']
True ["?'i_S", '?i_S']
False ['m_S', '?m_S', 'm_B', '?m_B']
False ['z_E', 's_E', '?s_E']
True ["?'i_I", '?i_I']
False ['m_I', '?m_I']
False ['?@-_E', '@-_E', "'@-_E"]
True ["'?o_B", '?o_B']
True ['?O_I', "'?O_I"]
True ['?A_B', "'?A_B"]
True ["?'E_E", "'?E_E", '?E_E']
True ["'?e_I", '?e_I']
False ['s_I', 'z_I', '?s_I']
True ["'?V_S", '?V_S']
True ["'?&+_S", '?&+_S']
```

```
True ['?&+_I', "'?&+_I"]
True ["'?W_B", '?W_B']
True ["?'i_E", '?i_E']
True ["'?W+_B", '?W+_B']
True ["?'E_I", "'?E_I", '?E_I']
True ["'?W_I", '?W_I']
True ["'?E_B", "?'E_B", '?E_B']
True ['?A_E', "'?A_E"]
True ['?&_E', "'?&_E"]
True ['?o_S', "'?o_S"]
True ["'?&_B", '?&_B']
True ['?W_E', "'?W_S", "'?W_E", '?W_S']
True ['?W+_S', '?W+_E', "'?W+_S", "'?W+_E"]
True ['?V_B', "'?V_B"]
True ['?&_S', "'?&_S"]
False ['?d_B', 'd_B']
False ['z_B', '?s_B', 's_B']
True ["'?e_B", '?e_B']
True ["'?e_E", '?e_E']
True ['?V_E', "'?V_E"]
True ['?A_I', "'?A_I", '?A_S', "'?A_S"]
True ['?O_E', "'?O_E"]
True ['?W+_I', "'?W+_I"]
False ["'?u_B", '?u_B', 'u_B', "'u_B"]
False ['u_E', '?u_E', "'?u_E", "'u_E"]
True ["'?&_I", '?&_I']
False ['?@-_I', "'@-_B", '?@-_B', '?@-_S', "'@-_S", '@-_I', '@-_S', '@-_B', "'@-_I"]
False ['?d_S', 'd_E', 'd_I', '?d_I', '?d_E', 'd_S']
True ["'?E_S", '?E_S', "?'E_S"]
True ['?e_S', "'?e_S"]
True ["'?y_I", '?y_I']
True ['?O_B', "'?O_B"]
True ["'?&+_E", '?&+_E']
True ['?i_B', "?'i_B"]
True ['?o_E', "'?o_E"]
True ["'?o_I", '?o_I']
False ['?u_S', "'?u_S", 'u_S', "'u_S"]
True ["'?y_E", "'?y_B", '?y_E', "'?y_S", '?y_S', '?y_B']
True ['?&+_B', "'?&+_B"]
```

### B.4.2 PLP+stød

#### B.4.2.1 tri4b

```
True ['?V_E', "'?V_E"]
True ['?V_B', "'?V_B"]
True ['?u_S', "'?u_S"]
True ["'?y_E", "'?y_B", '?y_E', "'?y_S", '?y_S', '?y_B']
```

```
False ['m_B', '?m_B']
True ['?A_E', "'?A_E"]
True ["'?o_I", '?o_I']
True ["'?W_I", '?W_I']
True ['?W+_S', '?W+_E', "'?W+_S", "'?W+_E"]
True ['?&_E', "'?&_E"]
True ["'?y_I", '?y_I']
True ["?'i_E", '?i_E']
True ['?W_E', "'?W_S", "'?W_E", '?W_S']
True ["'?V_I", '?V_I']
True ['?&+_I', "'?&+_I"]
True ['?i_B', "?'i_B"]
True ['?O_I', "'?O_I"]
True ['?W+_I', "'?W+_I"]
True ['?O_E', "'?O_E"]
True ['?e_S', "'?e_S"]
True ["'?&_I", '?&_I']
True ["'?W+_B", '?W+_B']
True ["'?u_B", '?u_B']
True ['?O_B', "'?O_B"]
True ["'?E_B", "?'E_B", '?E_B']
False ['z_E', 's_E', '?s_E']
False ['?d_S', 'd_E', 'd_I', '?d_I', '?d_E', 'd_S']
False ['z_B', '?s_B', 's_B']
True ['?O_S', "'?O_S"]
True ["?'E_I", "'?E_I", '?E_I']
True ["'?e_I", '?e_I']
False ['&_S', "'&_S", '?&_S', "'?&_S"]
True ['?A_I', "'?A_I", '?A_S', "'?A_S"]
True ["?'i_S", '?i_S']
True ["'?o_B", '?o_B']
False ['?d_B', 'd_B']
True ["'?V_S", '?V_S']
True ['?&+_B', "'?&+_B"]
True ['?u_I', "'?u_I"]
True ["'?W_B", '?W_B']
True ['?A_B', "'?A_B"]
True ["?'i_I", '?i_I']
False ['?@-_I', "'@-_B", '?@-_B', '?@-_S', "'@-_S", '@-_I', '@-_S', '@-_B', "'@-_I"]
False ['?m_E', 'm_E']
True ['?u_E', "'?u_E"]
False ['m_I', '?m_I']
False ['?O_I', 'O_I', 'O_S', '?O_S', "'O_E", '?O_B', "'O_B", "'O_I", 'O_B', "'?O_E",
"'?O_B", "'?O_S", 'O_E', "'O_S", '?O_E', "'?O_I"]
False ["'?&+_S", '?&+_S', "'&+_S", '&+_S']
True ['?o_S', "'?o_S"]
True ["'?E_S", '?E_S', "?'E_S"]
True ['?o_E', "'?o_E"]
```

```
True ["'?e_B", '?e_B']
True ["'?e_E", '?e_E']
True ["?'E_E", "'?E_E", '?E_E']
False ['?@-_E', '@-_E', "'@-_E"]
False ['s_I', 'z_I', '?s_I']
False ['s_S', '?s_S', 'z_S']
True ["'?&+_E", '?&+_E']
True ["'?&_B", '?&_B']
False ['m_S', '?m_S']
```

#### B.4.2.2 nnet5c

```
True ['?V_E', "'?V_E"]
True ['?V_B', "'?V_B"]
True ['?u_S', "'?u_S"]
True ["'?y_E", "'?y_B", '?y_E', "'?y_S", '?y_S', '?y_B']
False ['m_B', '?m_B']
True ['?A_E', "'?A_E"]
True ["'?o_I", '?o_I']
True ["'?W_I", '?W_I']
True ['?W+_S', '?W+_E', "'?W+_S", "'?W+_E"]
True ['?&_E', "'?&_E"]
True ["'?y_I", '?y_I']
True ["?'i_E", '?i_E']
True ['?W_E', "'?W_S", "'?W_E", '?W_S']
True ["'?V_I", '?V_I']
True ['?&+_I', "'?&+_I"]
True ['?i_B', "?'i_B"]
True ['?O_I', "'?O_I"]
True ['?W+_I', "'?W+_I"]
True ['?O_E', "'?O_E"]
True ['?e_S', "'?e_S"]
True ["'?&_I", '?&_I']
True ["'?W+_B", '?W+_B']
True ["'?u_B", '?u_B']
True ['?O_B', "'?O_B"]
True ["'?E_B", "?'E_B", '?E_B']
False ['z_E', 's_E', '?s_E']
False ['?d_S', 'd_E', 'd_I', '?d_I', '?d_E', 'd_S']
False ['z_B', '?s_B', 's_B']
True ['?O_S', "'?O_S"]
True ["?'E_I", "'?E_I", '?E_I']
True ["'?e_I", '?e_I']
False ['&_S', "'&_S", '?&_S', "'?&_S"]
True ['?A_I', "'?A_I", '?A_S', "'?A_S"]
True ["?'i_S", '?i_S']
True ["'?o_B", '?o_B']
False ['?d_B', 'd_B']
```

```
True ["'?V_S", '?V_S']
True ['?&+_B', "'?&+_B"]
True ['?u_I', "'?u_I"]
True ["'?W_B", '?W_B']
True ['?A_B', "'?A_B"]
True ["?'i_I", '?i_I']
False ['?@-_I', "'@-_B", '?@-_B', '?@-_S', "'@-_S", '@-_I', '@-_S', '@-_B', "'@-_I"]
False ['?m_E', 'm_E']
True ['?u_E', "'?u_E"]
False ['m_I', '?m_I']
False ['?O_I', 'O_I', 'O_S', '?O_S', "'O_E", '?O_B', "'O_B", "'O_I", 'O_B', "'?O_E", "'?O_B",
"'?O_S", 'O_E', "'O_S", '?O_E', "'?O_I"]
False ["'?&+_S", '?&+_S', "'&+_S", '&+_S']
True ['?o_S', "'?o_S"]
True ["'?E_S", '?E_S', "?'E_S"]
True ['?o_E', "'?o_E"]
True ["'?e_B", '?e_B']
True ["'?e_E", '?e_E']
True ["?'E_E", "'?E_E", '?E_E']
False ['?@-_E', '@-_E', "'@-_E"]
False ['s_I', 'z_I', '?s_I']
False ['s_S', '?s_S', 'z_S']
True ["'?&+_E", '?&+_E']
True ["'?&_B", '?&_B']
False ['m_S', '?m_S']
```

### B.4.3 MFCC+stød+pitch

#### B.4.3.1 nnet5c

```
False ['m_B', '?m_B']
False ['?u_I', 'u_I', "'u_I", "'?u_I"]
True ["'?o_I", '?o_I']
True ['?O_S', "'?O_S"]
True ['?A_E', '?A_I', "'?A_I", "'?A_E", '?A_S', "'?A_S"]
False ['s_S', '?s_S', 'z_S']
True ["'?V_I", '?V_I']
True ['?O_E', "'?O_E"]
True ["'?e_I", '?e_I']
True ["'?V_S", '?V_S']
False ["'?&+_S", '?&+_S', "'&+_S", '&+_S']
True ['?V_E', "'?V_E"]
True ['?&_E', "'?&_E"]
True ["'?y_B", "'?y_S", '?y_S', '?y_B']
False ['?@-_E', '@-_E', "'@-_E"]
True ['?o_E', "'?o_E"]
True ["?'i_E", '?i_E']
```

```
True ["'?o_B", '?o_B']
False ['?O_I', 'O_I', 'O_S', '?O_S', "'O_E", '?O_B', "'O_B", "'O_I", 'O_B', "'?O_E",
"'?O_B", "'?O_S", 'O_E', "'O_S", '?O_E', "'?O_I"]
True ["'?W_I", '?W_I']
False ['s_I', 'z_I', '?s_I']
True ['?&+_I', "'?&+_I"]
True ["'?y_E", '?y_E']
True ["?'i_I", '?i_I']
False ['?m_E', 'm_E']
True ['?o_S', "'?o_S"]
True ['?&_S', "'?&_S"]
True ["'?W_B", '?W_B']
True ["'?y_I", '?y_I']
True ["'?&_I", '?&_I']
True ['?W_E', "'?W_E"]
True ['?W+_I', "'?W+_I"]
True ["'?e_B", '?e_B']
False ["'?u_B", '?u_B', 'u_B', "'u_B"]
True ['?W+_S', "'?W+_S"]
True ["'?&+_E", '?&+_E']
True ["'?W+_B", '?W+_B']
False ['m_I', '?m_I']
True ["?'E_I", "'?E_I", '?E_I']
True ['?V_B', "'?V_B"]
True ['?e_S', "'?e_S"]
False ['z_B', '?s_B', 's_B']
True ["'?E_S", '?E_S', "?'E_S"]
True ["'?E_B", "?'E_B", '?E_B']
True ['?W+_E', "'?W+_E"]
True ["'?e_E", '?e_E']
False ['?u_S', "'?u_S", 'u_S', "'u_S"]
True ['?O_I', "'?O_I"]
True ["?'E_E", "'?E_E", '?E_E']
False ['?@-_I', "'@-_B", '?@-_B', '?@-_S', "'@-_S", '@-_I', '@-_S', '@-_B', "'@-_I"]
False ['u_E', '?u_E', "'?u_E", "'u_E"]
False ['?d_B', 'd_B']
False ['?d_S', 'd_E', 'd_I', '?d_I', '?d_E', 'd_S']
False ['m_S', '?m_S']
True ["?'i_S", '?i_S']
False ['z_E', 's_E', '?s_E']
True ["'?&_B", '?&_B']
True ['?&+_B', "'?&+_B"]
True ["'?W_S", '?W_S']
True ['?i_B', "?'i_B"]
False ['?A_B', "'?A_B", 'A_B', "'A_B"]
True ['?O_B', "'?O_B"]
```

### B.4.3.2 tri4b

```
False ['m_B', '?m_B']
False ['?u_I', 'u_I', "'u_I", "'?u_I"]
True ["'?o_I", '?o_I']
True ['?O_S', "'?O_S"]
True ['?A_E', '?A_I', "'?A_I", "'?A_E", '?A_S', "'?A_S"]
False ['s_S', '?s_S', 'z_S']
True ["'?V_I", '?V_I']
True ['?O_E', "'?O_E"]
True ["'?e_I", '?e_I']
True ["'?V_S", '?V_S']
False ["'?&+_S", '?&+_S', "'&+_S", '&+_S']
True ['?V_E', "'?V_E"]
True ['?&_E', "'?&_E"]
True ["'?y_B", "'?y_S", '?y_S', '?y_B']
False ['?@-_E', '@-_E', "'@-_E"]
True ['?o_E', "'?o_E"]
True ["?'i_E", '?i_E']
True ["'?o_B", '?o_B']
False ['?O_I', 'O_I', 'O_S', '?O_S', "'O_E", '?O_B', "'O_B", "'O_I", 'O_B', "'?O_E",
"'?O_B", "'?O_S", 'O_E', "'O_S", '?O_E', "'?O_I"]
True ["'?W_I", '?W_I']
False ['s_I', 'z_I', '?s_I']
True ['?&+_I', "'?&+_I"]
True ["'?y_E", '?y_E']
True ["?'i_I", '?i_I']
False ['?m_E', 'm_E']
True ['?o_S', "'?o_S"]
True ['?&_S', "'?&_S"]
True ["'?W_B", '?W_B']
True ["'?y_I", '?y_I']
True ["'?&_I", '?&_I']
True ['?W_E', "'?W_E"]
True ['?W+_I', "'?W+_I"]
True ["'?e_B", '?e_B']
False ["'?u_B", '?u_B', 'u_B', "'u_B"]
True ['?W+_S', "'?W+_S"]
True ["'?&+_E", '?&+_E']
True ["'?W+_B", '?W+_B']
False ['m_I', '?m_I']
True ["?'E_I", "'?E_I", '?E_I']
True ['?V_B', "'?V_B"]
True ['?e_S', "'?e_S"]
False ['z_B', '?s_B', 's_B']
True ["'?E_S", '?E_S', "?'E_S"]
True ["'?E_B", "?'E_B", '?E_B']
True ['?W+_E', "'?W+_E"]
```

```
True ["'?e_E", '?e_E']
False ['?u_S', "'?u_S", 'u_S', "'u_S"]
True ['?O_I', "'?O_I"]
True ["?'E_E", "'?E_E", '?E_E']
False ['?@-_I', "'@-_B", '?@-_B', '?@-_S', "'@-_S", '@-_I', '@-_S', '@-_B', "'@-_I"]
False ['u_E', '?u_E', "'?u_E", "'u_E"]
False ['?d_B', 'd_B']
False ['?d_S', 'd_E', 'd_I', '?d_I', '?d_E', 'd_S']
False ['m_S', '?m_S']
True ["?'i_S", '?i_S']
False ['z_E', 's_E', '?s_E']
True ["'?&_B", '?&_B']
True ['?&+_B', "'?&+_B"]
True ["'?W_S", '?W_S']
True ['?i_B', "?'i_B"]
False ['?A_B', "'?A_B", 'A_B', "'A_B"]
True ['?O_B', "'?O_B"]
```

### B.4.4 PLP+stød+pitch

#### B.4.4.1 tri4b

```
True ["?'i_E", '?i_E']
False ['?A_E', "'?A_E", 'A_E', "'A_E"]
False ['?o_S', "'?o_S", "'o_S", 'o_S']
False ['?d_S', 'd_E', '?d_E', 'd_S']
True ["'?y_I", '?y_I']
True ['?O_S', "'?O_S"]
True ['?i_B', "?'i_B"]
True ["'?e_B", '?e_B']
False ['?m_E', 'm_E']
True ["'?e_I", '?e_I']
True ["?'i_I", '?i_I']
True ["'?&_B", '?&_B']
True ["?'i_S", '?i_S']
True ["'?&+_E", '?&+_E']
True ['?&+_I', "'?&+_I"]
True ["?'E_I", "'?E_I", '?E_I']
True ['?O_B', "'?O_B"]
False ['m_I', 'm_S', '?m_S', '?m_I']
True ["'?E_B", "?'E_B", '?E_B']
False ["'o_E", 'o_E', '?o_E', "'?o_E"]
True ['?&_E', "'?&_E"]
False ['s_I', 'z_I', '?s_I']
False ['z_B', '?s_B', 's_B']
True ["'?e_E", '?e_E']
False ['?A_B', "'?A_B", 'A_B', "'A_B"]
False ['s_S', '?s_S', 'z_S']
```

```
False ['m_B', '?m_B']
True ["'?V_S", '?V_S']
True ['?W+_I', "'?W+_I", "'?W_I", '?W_I']
True ["'?E_S", '?E_S', "?'E_E", "'?E_E", '?E_E', "?'E_S"]
False ['?u_S', "'?u_S", 'u_S', "'u_S"]
False ['z_E', 's_E', '?s_E']
True ["'?y_E", "'?y_B", '?y_E', "'?y_S", '?y_S', '?y_B']
True ["'?V_I", '?V_I']
True ['?V_E', "'?V_E"]
False ['?@-_E', '@-_E', "'@-_E"]
False ['u_E', '?u_E', "'?u_E", "'u_E"]
True ["'?&+_S", '?&+_S']
True ["'?W_B", '?W_B']
False ['?@-_I', "'@-_B", '?@-_B', '?@-_S', "'@-_S", '@-_I', '@-_S', '@-_B', "'@-_I"]
False ['?u_I', 'u_I', "'u_I", "'?u_I"]
False ['?O_I', 'O_I', 'O_S', '?O_S', "'O_E", '?O_B', "'O_B", "'O_I", 'O_B', "'?O_E",
"'?O_B", "'?O_S", 'O_E', "'O_S", '?O_E', "'?O_I"]
False ['?d_B', 'd_B']
True ['?W+_S', '?W_E', "'?W_S", '?W+_E', "'?W_E", "'?W+_S", "'?W+_E", '?W_S']
False ['d_I', '?d_I']
True ['?e_S', "'?e_S"]
True ["'?&_I", '?&_I']
True ["'?W+_B", '?W+_B']
True ['?&_S', "'?&_S"]
True ['?O_E', "'?O_E"]
True ['?&+_B', "'?&+_B"]
False ['?A_I', "'?A_I", 'A_I', "'A_I"]
True ["'?o_I", '?o_I']
False ["'?o_B", 'o_B', '?o_B', "'o_B"]
False ["'?u_B", '?u_B', 'u_B', "'u_B"]
True ['?V_B', "'?V_B"]
False ['A_S', '?A_S', "'A_S", "'?A_S"]
True ['?O_I', "'?O_I"]
```

#### B.4.4.2 nnet5c

```
True ["?'i_E", '?i_E']
False ['?A_E', "'?A_E", 'A_E', "'A_E"]
False ['?o_S', "'?o_S", "'o_S", 'o_S']
False ['?d_S', 'd_E', '?d_E', 'd_S']
True ["'?y_I", '?y_I']
True ['?O_S', "'?O_S"]
True ['?i_B', "?'i_B"]
True ["'?e_B", '?e_B']
False ['?m_E', 'm_E']
True ["'?e_I", '?e_I']
True ["?'i_I", '?i_I']
```

```
True ["'?&_B", '?&_B']
True ["?'i_S", '?i_S']
True ["'?&+_E", '?&+_E']
True ['?&+_I', "'?&+_I"]
True ["?'E_I", "'?E_I", '?E_I']
True ['?O_B', "'?O_B"]
False ['m_I', 'm_S', '?m_S', '?m_I']
True ["'?E_B", "?'E_B", '?E_B']
False ["'o_E", 'o_E', '?o_E', "'?o_E"]
True ['?&_E', "'?&_E"]
False ['s_I', 'z_I', '?s_I']
False ['z_B', '?s_B', 's_B']
True ["'?e_E", '?e_E']
False ['?A_B', "'?A_B", 'A_B', "'A_B"]
False ['s_S', '?s_S', 'z_S']
False ['m_B', '?m_B']
True ["'?V_S", '?V_S']
True ['?W+_I', "'?W+_I", "'?W_I", '?W_I']
True ["'?E_S", '?E_S', "?'E_E", "'?E_E", '?E_E', "?'E_S"]
False ['?u_S', "'?u_S", 'u_S', "'u_S"]
False ['z_E', 's_E', '?s_E']
True ["'?y_E", "'?y_B", '?y_E', "'?y_S", '?y_S', '?y_B']
True ["'?V_I", '?V_I']
True ['?V_E', "'?V_E"]
False ['?@-_E', '@-_E', "'@-_E"]
False ['u_E', '?u_E', "'?u_E", "'u_E"]
True ["'?&+_S", '?&+_S']
True ["'?W_B", '?W_B']
False ['?@-_I', "'@-_B", '?@-_B', '?@-_S', "'@-_S", '@-_I', '@-_S', '@-_B', "'@-_I"]
False ['?u_I', 'u_I', "'u_I", "'?u_I"]
False ['?O_I', 'O_I', 'O_S', '?O_S', "'O_E", '?O_B', "'O_B", "'O_I", 'O_B', "'?O_E", "'?O_B",
"'?O_S", 'O_E', "'O_S", '?O_E', "'?O_I"]
False ['?d_B', 'd_B']
True ['?W+_S', '?W_E', "'?W_S", '?W+_E', "'?W_E", "'?W+_S", "'?W+_E", '?W_S']
False ['d_I', '?d_I']
True ['?e_S', "'?e_S"]
True ["'?&_I", '?&_I']
True ["'?W+_B", '?W+_B']
True ['?&_S', "'?&_S"]
True ['?O_E', "'?O_E"]
True ['?&+_B', "'?&+_B"]
False ['?A_I', "'?A_I", 'A_I', "'A_I"]
True ["'?o_I", '?o_I']
False ["'?o_B", 'o_B', '?o_B', "'o_B"]
False ["'?u_B", '?u_B', 'u_B', "'u_B"]
True ['?V_B', "'?V_B"]
False ['A_S', '?A_S', "'A_S", "'?A_S"]
True ['?O_I', "'?O_I"]
```

### B.4.5 Extended feature sets

This section contain equivalence classes from the nnet5c systems trained in Chapter 6. The equivalence classes that are in common between nnet5c systems used in Chapter 6 are in Section B.4.5.3.

#### B.4.5.1 MFCC+pitch+HRF

```
False ['?m_I', 'm_I']
True ['?u_I', "'?u_I"]
True ['?i_S', "?'i_S"]
True ['?y_B', "'?y_B"]
True ["'?y_I", '?y_I']
False ['z_S', 's_S', '?s_S']
False ['?s_B', 's_B', 'z_B']
True ["'?E_S", "?'E_S", '?E_S']
True ["'?u_B", '?u_B']
False ['m_S', '?m_B', '?m_S', 'm_B']
True ["'?e_I", '?e_I']
True ['?&_S', "'?&+_S", '?&+_S', "'?&_S"]
True ['?o_E', "'?o_E"]
True ["'?W_B", '?W_B']
False ['@_B', "'@-_I", '@-_S', '?@-_S', "'@-_S", '@_I', '@-_I', "'@_B", "'@_S",
'?@-_B', '?@-_I', '@_S', "'@-_B", "'@_I", '@-_B']
True ['?e_B', "'?e_B"]
True ['?&+_I', "'?&+_I"]
False ['d_B', '?d_B']
True ["'?&_I", '?&_I']
False ['?s_E', 's_E', 'z_E']
True ['?u_E', "'?u_E"]
True ['?o_I', "'?o_I"]
True ['?o_B', "'?o_B"]
True ["'?e_E", '?e_E']
True ["?'E_E", '?E_E', "'?E_E"]
True ["'?O_S", '?O_S']
True ["?'i_E", '?i_E']
True ['?&_B', "'?&_B"]
True ["'?o_S", '?o_S']
True ['?i_B', "?'i_B"]
True ['?O_I', "'?O_I"]
True ["'?W+_S", "'?W_E", "'?W+_E", '?W_S', '?W_E', '?W+_S', '?W+_E', "'?W_S"]
False ['?@-_E', '@-_E', "'@-_E"]
True ['?i_I', "?'i_I"]
True ["?'E_I", '?E_I', "'?E_I"]
True ['?V_I', "'?V_I"]
True ["'?O_E", '?O_E']
```

```
False ['z_I', 's_I', '?s_I']
False ["'?O_E", 'O_E', 'O_S', "'?O_B", '?O_B', "'O_I", "'O_B", "'O_E", 'O_B',
"'O_S", '?O_I', "'?O_S", '?O_E', 'O_I', '?O_S', "'?O_I"]
True ["'?y_E", '?y_E']
True ["'?V_B", '?V_B']
True ['?&+_B', "'?&+_B"]
True ["'?e_S", '?e_S']
True ["'?E_B", '?E_B', "?'E_B"]
True ["'?V_E", '?V_E']
True ["'?A_E", '?A_S', '?A_I', '?A_E', "'?A_I", "'?A_S"]
True ['?y_S', "'?y_S"]
True ["'?u_S", '?u_S']
True ["'?W_I", '?W_I']
True ['?&_E', "'?&_E"]
True ['?W+_B', "'?W+_B"]
False ['m_E', '?m_E']
True ['?V_S', "'?V_S"]
True ["'?W+_I", '?W+_I']
True ['?&+_E', "'?&+_E"]
False ['A_B', "'A_B", '?A_B', "'?A_B"]
True ['?O_B', "'?O_B"]
False ['d_E', 'd_I', '?d_S', 'd_S', '?d_E', '?d_I']
```

#### B.4.5.2 MFCC+pitch+phase

```
True ["'?V_E", '?V_E']
True ['?i_I', "?'i_I"]
True ["'?W_I", '?W_I']
True ['?e_B', "'?e_B"]
False ['m_S', '?m_S']
False ['d_B', '?d_B']
True ["'?O_E", '?O_E']
False ["'&+_E", '&+_E', "'&_E", '&_E', '?&+_E', '?&_E', "'?&_E", "'?&+_E"]
True ['?O_I', "'?O_I"]
True ['?o_I', "'?o_I"]
True ["'?E_S", "?'E_S", '?E_S']
False ['?m_I', 'm_I']
True ["'?W+_S", "'?W+_E", '?W+_S', '?W+_E']
True ["'?e_I", '?e_I']
False ['?s_E', 's_E', 'z_E']
True ['?u_E', "'?u_E"]
False ["'?O_E", 'O_E', 'O_S', "'?O_B", '?O_B', "'O_I", "'O_B", "'O_E", 'O_B',
"'O_S", '?O_I', "'?O_S", '?O_E', 'O_I', '?O_S', "'?O_I"]
True ["'?O_S", '?O_S']
True ["'?e_E", '?e_E']
False ['?@-_E', '@-_E', "'@-_E"]
True ["'?V_B", '?V_B']
True ["'?u_B", '?u_B']
```

```
True ["'?W_B", '?W_B']
False ['m_E', '?m_E']
False ['d_E', 'd_I', '?d_S', 'd_S', '?d_E', '?d_I']
True ["'?u_S", '?u_S']
True ["?'i_E", '?i_E']
False ["'@-_I", '@-_S', '?@-_S', "'@-_S", '@-_I', '?@-_B', '?@-_I', "'@-_B", '@-_B']
True ["'?W_E", '?W_S', '?W_E', "'?W_S"]
False ['z_I', 's_I', '?s_I']
True ['?V_I', "'?V_I"]
True ['?u_I', "'?u_I"]
False ['z_S', 's_S', '?s_S']
True ["'?W+_I", '?W+_I']
True ["'?e_S", '?e_S']
True ['?W+_B', "'?W+_B"]
True ['?V_S', "'?V_S"]
True ["'?E_B", '?E_B', "?'E_B"]
True ["?'E_I", '?E_I', "'?E_I"]
True ["'?y_I", '?y_I']
False ["'?A_E", 'A_E', '?A_E', "'A_E"]
True ['?y_B', "'?y_B", '?y_S', "'?y_S", "'?y_E", '?y_E']
True ['?o_E', "'?o_E"]
False ['A_I', '?A_S', "'A_I", '?A_I', "'A_S", "'?A_I", "'?A_S", 'A_S']
False ['o_B', "'o_B", '?o_B', "'?o_B"]
False ['?s_B', 's_B', 'z_B']
True ["'?o_S", '?o_S']
True ['?&+_I', "'?&+_I", "'?&_I", '?&_I']
False ['A_B', "'A_B", '?A_B', "'?A_B"]
True ['?i_S', "?'i_S"]
False ['?m_B', 'm_B']
True ['?O_B', "'?O_B"]
True ['?i_B', "?'i_B"]
True ["?'E_E", '?E_E', "'?E_E"]
True ['?&_S', "'?&+_S", '?&_B', "'?&_B", '?&+_S', '?&+_B', "'?&_S", "'?&+_B"]
```

### B.4.5.3 Common equivalence classes

```
False ["'?O_E", 'O_E', 'O_S', "'?O_B", '?O_B', "'O_I", "'O_B", "'O_E", 'O_B', "'O_S", '?O_I', "'?O_S", '?O_E', 'O_I', '?
False ['d_B', '?d_B']
False ['d_E', 'd_I', '?d_S', 'd_S', '?d_E', '?d_I']
False ['?@-_E', '@-_E', "'@-_E"]
False ['m_E', '?m_E']
False ['?m_I', 'm_I']
False ['?s_B', 's_B', 'z_B']
False ['?s_E', 's_E', 'z_E']
False ['z_I', 's_I', '?s_I']
False ['z_S', 's_S', '?s_S']
True ['?e_B', "'?e_B"]
True ["'?E_B", '?E_B', "?'E_B"]
```

```
True ["'?e_E", '?e_E']
True ["?'E_E", '?E_E', "'?E_E"]
True ["'?e_I", '?e_I']
True ["?'E_I", '?E_I', "'?E_I"]
True ["'?e_S", '?e_S']
True ["'?E_S", "?'E_S", '?E_S']
True ['?i_B', "?'i_B"]
True ["?'i_E", '?i_E']
True ['?i_I', "?'i_I"]
True ['?i_S', "?'i_S"]
True ['?O_B', "'?O_B"]
True ['?o_E', "'?o_E"]
True ["'?O_E", '?O_E']
True ['?o_I', "'?o_I"]
True ['?O_I', "'?O_I"]
True ["'?o_S", '?o_S']
True ["'?O_S", '?O_S']
True ["'?V_B", '?V_B']
True ["'?V_E", '?V_E']
True ['?V_I', "'?V_I"]
True ['?V_S', "'?V_S"]
True ['?W+_B', "'?W+_B"]
True ["'?W_B", '?W_B']
True ["'?W_I", '?W_I']
True ["'?W+_I", '?W+_I']
True ["'?y_I", '?y_I']
```

## B.5 Recognition Errors in Stasjon06

### B.5.1 MFCC+stød

```
CONFUSION PAIRS                 Total                 (3207)
                                With >=  1 occurances (3207)

   1:   36  ->  a/s ==> s
   2:   32  ->  o ==> u
   3:   31  ->  det ==> de
   4:   30  ->  af ==> er
   5:   24  ->  i ==> e
   6:   23  ->  fÅr ==> for
   7:   21  ->  er ==> af
   8:   19  ->  Åbn ==> Åben
   9:   18  ->  af ==> at
  10:   18  ->  gÅ ==> gÅr


INSERTIONS                      Total                 (869)
                                With >=  1 occurances (869)

   1:  502  ->  punktum
   2:   70  ->  i
   3:   54  ->  og
   4:   44  ->  e
   5:   44  ->  en
   6:   34  ->  at
   7:   32  ->  til
   8:   29  ->  n
   9:   28  ->  er
  10:   24  ->  for
```

```
DELETIONS                       Total                 (261)
                                With >=  1 occurances (261)

   1:  519  ->  punktum
   2:  109  ->  i
   3:  103  ->  er
   4:   66  ->  og
   5:   56  ->  at
   6:   41  ->  et
   7:   32  ->  af
   8:   28  ->  det
   9:   25  ->  der
  10:   23  ->  har


SUBSTITUTIONS                   Total                 (1942)
                                With >=  1 occurances (1942)

   1:   88  ->  af
   2:   87  ->  er
   3:   83  ->  punktum
   4:   79  ->  og
   5:   74  ->  i
   6:   64  ->  det
   7:   49  ->  a/s
   8:   44  ->  en
   9:   40  ->  at
  10:   39  ->  et
```

### B.5.2 MFCC+stød+pitch

```
CONFUSION PAIRS                 Total                 (3207)
                                With >=  1 occurances (3207)

   1:   36  ->  a/s ==> s
   2:   32  ->  o ==> u
   3:   31  ->  det ==> de
   4:   30  ->  af ==> er
   5:   24  ->  i ==> e
   6:   23  ->  fÅr ==> for
   7:   21  ->  er ==> af
   8:   19  ->  Åbn ==> Åben
   9:   18  ->  af ==> at
  10:   18  ->  gÅ ==> gÅr


INSERTIONS                      Total                 (869)
                                With >=  1 occurances (869)

   1:  502  ->  punktum
   2:   70  ->  i
   3:   54  ->  og
   4:   44  ->  e
   5:   44  ->  en
   6:   34  ->  at
   7:   32  ->  til
   8:   29  ->  n
   9:   28  ->  er
  10:   24  ->  for


DELETIONS                       Total                 (261)
```

```
                                With >=  1 occurances (261)

   1:  519  ->  punktum
   2:  109  ->  i
   3:  103  ->  er
   4:   66  ->  og
   5:   56  ->  at
   6:   41  ->  et
   7:   32  ->  af
   8:   28  ->  det
   9:   25  ->  der
  10:   23  ->  har


SUBSTITUTIONS                   Total                 (1942)
                                With >=  1 occurances (1942)

   1:   88  ->  af
   2:   87  ->  er
   3:   83  ->  punktum
   4:   79  ->  og
   5:   74  ->  i
   6:   64  ->  det
   7:   49  ->  a/s
   8:   44  ->  en
   9:   40  ->  at
  10:   39  ->  et
```

### B.5.3 CASE: punktum

#### B.5.3.1 Confusion pairs

```
 27:   11  ->  e ==> punktum
 28:   10  ->  a/s ==> punktum
 42:    8  ->  brevenstre ==> punktum
133:    6  ->  modregning ==> punktum
135:    6  ->  naturlige ==> punktum
163:    5  ->  baggrundsfarvenstre ==> punktum
164:    5  ->  billiardkuglen ==> punktum
196:    5  ->  negativenstre ==> punktum
265:    4  ->  punktum ==> ville
351:    3  ->  ind ==> punktum
354:    3  ->  jer ==> punktum
370:    3  ->  punktum ==> bakken
371:    3  ->  punktum ==> forelskelse
372:    3  ->  punktum ==> gymnastik
373:    3  ->  punktum ==> skal
374:    3  ->  r ==> punktum
420:    2  ->  akhtar ==> punktum
431:    2  ->  bachmann ==> punktum
432:    2  ->  bag ==> punktum
456:    2  ->  dag ==> punktum
467:    2  ->  den ==> punktum
471:    2  ->  der ==> punktum
538:    2  ->  hØjlyng ==> punktum
579:    2  ->  mejer ==> punktum
580:    2  ->  midtgaard ==> punktum
586:    2  ->  mosegaard ==> punktum
614:    2  ->  punktum ==> format
615:    2  ->  punktum ==> multimedier
616:    2  ->  punktum ==> mÅde
617:    2  ->  punktum ==> mÆnd
```

```
640:    2  ->  store ==> punktum
650:    2  ->  sÅ ==> punktum
663:    2  ->  valentin ==> punktum
669:    2  ->  vestermark ==> punktum
670:    2  ->  vi ==> punktum
760:    1  ->  allingÅlyst ==> punktum
765:    1  ->  alofi ==> punktum
766:    1  ->  alsace ==> punktum
775:    1  ->  amazon ==> punktum
777:    1  ->  amstrup ==> punktum
778:    1  ->  amtskommune ==> punktum
779:    1  ->  amtssygehus ==> punktum
783:    1  ->  anden ==> punktum
789:    1  ->  andet ==> punktum
797:    1  ->  annapurne ==> punktum
799:    1  ->  antonio ==> punktum
804:    1  ->  aps ==> punktum
863:    1  ->  bak ==> punktum
865:    1  ->  ballerina ==> punktum
867:    1  ->  bangladesh ==> punktum
874:    1  ->  basse ==> punktum
875:    1  ->  bastemose ==> punktum
884:    1  ->  belastning ==> punktum
888:    1  ->  bengtson ==> punktum
889:    1  ->  beringshavet ==> punktum
893:    1  ->  bertram ==> punktum
894:    1  ->  besaturation ==> punktum
907:    1  ->  bi ==> punktum
908:    1  ->  bigum ==> punktum
910:    1  ->  bilde ==> punktum
915:    1  ->  billede ==> punktum
927:    1  ->  bjerreby ==> punktum
928:    1  ->  bjerring ==> punktum
```

```
 936:    1  ->  blivenstre ==> punktum
 939:    1  ->  bliver ==> punktum
 945:    1  ->  boel ==> punktum
 956:    1  ->  bredlÆnge ==> punktum
 960:    1  ->  broch ==> punktum
 964:    1  ->  broender ==> punktum
 973:    1  ->  bruge ==> punktum
 976:    1  ->  bruus ==> punktum
 979:    1  ->  brÅby ==> punktum
 981:    1  ->  brØdsgaard ==> punktum
 982:    1  ->  brØgger ==> punktum
 987:    1  ->  byskov ==> punktum
 991:    1  ->  bÆrentsen ==> punktum
 996:    1  ->  calella ==> punktum
 998:    1  ->  castries ==> punktum
 999:    1  ->  cateringudstyr ==> punktum
1018:    1  ->  cognacen ==> punktum
1027:    1  ->  d ==> punktum
1036:    1  ->  da ==> punktum
1056:    1  ->  dansbjerg ==> punktum
1074:    1  ->  de ==> punktum
1078:    1  ->  deleuran ==> punktum
1138:    1  ->  dig ==> punktum
1153:    1  ->  domfÆldelser ==> punktum
1155:    1  ->  drachmann ==> punktum
1164:    1  ->  drewsen ==> punktum
1167:    1  ->  drÅbydalen ==> punktum
1175:    1  ->  dÆmningen ==> punktum
1176:    1  ->  dØrup ==> punktum
1217:    1  ->  emborg ==> punktum
1258:    1  ->  engang ==> punktum
1259:    1  ->  enge ==> punktum
1264:    1  ->  enghaveskolen ==> punktum
```

```
1268:    1  ->  enorme ==> punktum
1298:    1  ->  eritrea ==> punktum
1300:    1  ->  ertebjergskov ==> punktum
1325:    1  ->  factory ==> punktum
1328:    1  ->  faksinge ==> punktum
1336:    1  ->  farup ==> punktum
1339:    1  ->  fastemose ==> punktum
1394:    1  ->  fordi ==> punktum
1406:    1  ->  forliget ==> punktum
1424:    1  ->  fortsatte ==> punktum
1453:    1  ->  fyldt ==> punktum
1461:    1  ->  fÆrdigudfØrt ==> punktum
1475:    1  ->  galapagosØerne ==> punktum
1476:    1  ->  galway ==> punktum
1477:    1  ->  gammelmark ==> punktum
1494:    1  ->  georgetown ==> punktum
1495:    1  ->  georgia ==> punktum
1514:    1  ->  glemslens ==> punktum
1526:    1  ->  gormsen ==> punktum
1531:    1  ->  gravgaard ==> punktum
1535:    1  ->  grØnkjÆr ==> punktum
1537:    1  ->  grØnlØkke ==> punktum
1540:    1  ->  guldager ==> punktum
1544:    1  ->  gustavsen ==> punktum
1551:    1  ->  gÅrde ==> punktum
1553:    1  ->  gÅrdstedgÅrd ==> punktum
1554:    1  ->  gÅrsdal ==> punktum
1559:    1  ->  hahn ==> punktum
1560:    1  ->  halkjÆr ==> punktum
1561:    1  ->  hallum ==> punktum
1562:    1  ->  halvorsen ==> punktum
1571:    1  ->  handi ==> punktum
1583:    1  ->  hartvig ==> punktum
```

```
1596:    1  ->  hav ==> punktum
1597:    1  ->  havbro ==> punktum
1607:    1  ->  hebron ==> punktum
1610:    1  ->  hellevad ==> punktum
1632:    1  ->  hesbjerg ==> punktum
1643:    1  ->  hildebrandt ==> punktum
1655:    1  ->  hobet ==> punktum
1659:    1  ->  holdning ==> punktum
1662:    1  ->  holskovbÆk ==> punktum
1667:    1  ->  houmann ==> punktum
1682:    1  ->  hussein ==> punktum
1701:    1  ->  hvordan ==> punktum
1713:    1  ->  hØegh ==> punktum
1714:    1  ->  hØeghsgÅrd ==> punktum
1715:    1  ->  hØier ==> punktum
1720:    1  ->  hØjtved ==> punktum
1746:    1  ->  ibsgÅrde ==> punktum
1803:    1  ->  ismail ==> punktum
1805:    1  ->  israel ==> punktum
1826:    1  ->  jernhatten ==> punktum
1841:    1  ->  juryen ==> punktum
1855:    1  ->  karlebjerg ==> punktum
1857:    1  ->  karrierejob ==> punktum
1864:    1  ->  katedralskole ==> punktum
1866:    1  ->  kelleklinte ==> punktum
1873:    1  ->  kioga ==> punktum
1876:    1  ->  kirgizistan ==> punktum
1878:    1  ->  kjeldhØjgÅrd ==> punktum
1898:    1  ->  kolding ==> punktum
1924:    1  ->  korea ==> punktum
1930:    1  ->  kostrÆde ==> punktum
1933:    1  ->  krampagtigt ==> punktum
1937:    1  ->  kregme ==> punktum
```

```
1944:    1  ->  kromutter ==> punktum
1963:    1  ->  kyed ==> punktum
1970:    1  ->  kØ ==> punktum
2000:    1  ->  lentz ==> punktum
2018:    1  ->  lillenor ==> punktum
2019:    1  ->  linde ==> punktum
2020:    1  ->  lindgren ==> punktum
2027:    1  ->  loft ==> punktum
2029:    1  ->  lomholt ==> punktum
2034:    1  ->  lucia ==> punktum
2038:    1  ->  lundgreen ==> punktum
2054:    1  ->  lyttesholm ==> punktum
2059:    1  ->  lÆrer ==> punktum
2080:    1  ->  malabo ==> punktum
2082:    1  ->  malling ==> punktum
2083:    1  ->  malta ==> punktum
2091:    1  ->  manaslu ==> punktum
2099:    1  ->  marcher ==> punktum
2138:    1  ->  mekong ==> punktum
2142:    1  ->  men ==> punktum
2183:    1  ->  monopolstilling ==> punktum
2184:    1  ->  monterrey ==> punktum
2188:    1  ->  moukÆr ==> punktum
2197:    1  ->  mÅ ==> punktum
2205:    1  ->  mØgelsig ==> punktum
2207:    1  ->  mØrup ==> punktum
2208:    1  ->  mØsthuse ==> punktum
2228:    1  ->  nevershuse ==> punktum
2229:    1  ->  nevis ==> punktum
2232:    1  ->  niemann ==> punktum
2233:    1  ->  nigeria ==> punktum
2235:    1  ->  nilen ==> punktum
2238:    1  ->  noget ==> punktum
```

```
2260:    1  ->  nÆvenstre ==> punktum
2323:    1  ->  opgavenstre ==> punktum
2348:    1  ->  os ==> punktum
2353:    1  ->  osman ==> punktum
2358:    1  ->  otto ==> punktum
2359:    1  ->  outzen ==> punktum
2365:    1  ->  overmotiverede ==> punktum
2380:    1  ->  pehrson ==> punktum
2386:    1  ->  personellet ==> punktum
2399:    1  ->  plagede ==> punktum
2403:    1  ->  poetisk ==> punktum
2411:    1  ->  positiv ==> punktum
2412:    1  ->  post ==> punktum
2421:    1  ->  private ==> punktum
2441:    1  ->  punktum ==> anelse
2442:    1  ->  punktum ==> atten
2443:    1  ->  punktum ==> autoresume
2444:    1  ->  punktum ==> autorisation
2445:    1  ->  punktum ==> beherskelse
2446:    1  ->  punktum ==> bibliotek
2447:    1  ->  punktum ==> boks
2448:    1  ->  punktum ==> broen
2449:    1  ->  punktum ==> conny
2450:    1  ->  punktum ==> datomÆrkningen
2451:    1  ->  punktum ==> debat
2452:    1  ->  punktum ==> det
2453:    1  ->  punktum ==> dokument
2454:    1  ->  punktum ==> fil
2455:    1  ->  punktum ==> filer
2456:    1  ->  punktum ==> flettet
2457:    1  ->  punktum ==> flytte
2458:    1  ->  punktum ==> fredet
2459:    1  ->  punktum ==> glÆden
```

```
2460:    1  ->  punktum ==> handlingen
2461:    1  ->  punktum ==> hjÆlpefunktion
2462:    1  ->  punktum ==> hold
2463:    1  ->  punktum ==> hovedet
2464:    1  ->  punktum ==> huse
2465:    1  ->  punktum ==> hver
2466:    1  ->  punktum ==> interesse
2467:    1  ->  punktum ==> kegler
2468:    1  ->  punktum ==> kravene
2469:    1  ->  punktum ==> krydse
2470:    1  ->  punktum ==> lukke
2471:    1  ->  punktum ==> man
2472:    1  ->  punktum ==> mange
2473:    1  ->  punktum ==> meldes
2474:    1  ->  punktum ==> menu
2475:    1  ->  punktum ==> mig
2476:    1  ->  punktum ==> militÆr
2477:    1  ->  punktum ==> modtageren
2478:    1  ->  punktum ==> muligt
2479:    1  ->  punktum ==> naboen
2480:    1  ->  punktum ==> ny
2481:    1  ->  punktum ==> nÆvn
2482:    1  ->  punktum ==> plakaten
2483:    1  ->  punktum ==> r
2484:    1  ->  punktum ==> rom
2485:    1  ->  punktum ==> se
2486:    1  ->  punktum ==> set
2487:    1  ->  punktum ==> show
2488:    1  ->  punktum ==> sted
2489:    1  ->  punktum ==> stikke
2490:    1  ->  punktum ==> stjerne
2491:    1  ->  punktum ==> styrt
2492:    1  ->  punktum ==> sÆsonen
```

```
2493:    1  ->  punktum ==> sÆtningen
2494:    1  ->  punktum ==> tidlØshed
2495:    1  ->  punktum ==> tilstrÆkkelige
2496:    1  ->  punktum ==> tilstrÆkkeligt
2497:    1  ->  punktum ==> vÆrd
2498:    1  ->  punktum ==> wc-kummer
2499:    1  ->  punktum ==> Ørsted
2510:    1  ->  pÆdagogikum ==> punktum
2522:    1  ->  rakte ==> punktum
2533:    1  ->  reerslev ==> punktum
2557:    1  ->  ringgaard ==> punktum
2574:    1  ->  rØnnebÆk ==> punktum
2600:    1  ->  se ==> punktum
2601:    1  ->  secher ==> punktum
2604:    1  ->  sejlstrup ==> punktum
2625:    1  ->  seychellerne ==> punktum
2626:    1  ->  sicilien ==> punktum
2636:    1  ->  sikret ==> punktum
2638:    1  ->  sildekule ==> punktum
2642:    1  ->  sjuskes ==> punktum
2646:    1  ->  skaaning ==> punktum
2659:    1  ->  skjØd ==> punktum
2661:    1  ->  skole ==> punktum
2665:    1  ->  skovmark ==> punktum
2678:    1  ->  skÅrupgÅrd ==> punktum
2695:    1  ->  slovenien ==> punktum
2704:    1  ->  socialrÅdgiverforening ==> punktum
2707:    1  ->  solbjerggÅrd ==> punktum
2714:    1  ->  sortenshave ==> punktum
2733:    1  ->  spØrger ==> punktum
2735:    1  ->  stadilby ==> punktum
2759:    1  ->  storehØj ==> punktum
2764:    1  ->  straarup ==> punktum
```

```
2767:    1  ->  strand ==> punktum
2768:    1  ->  stryhn ==> punktum
2787:    1  ->  sydpolen ==> punktum
2811:    1  ->  sØnderup ==> punktum
2828:    1  ->  tegnefilmsregi ==> punktum
2830:    1  ->  tekstbehandlingsprogrammet ==> punktum
2888:    1  ->  timeshare ==> punktum
2896:    1  ->  tofte ==> punktum
2899:    1  ->  tomÉ ==> punktum
2924:    1  ->  trolle ==> punktum
2925:    1  ->  truede ==> punktum
2927:    1  ->  trunbro ==> punktum
2946:    1  ->  tÅgeskov ==> punktum
2953:    1  ->  tØnding ==> punktum
2968:    1  ->  udgavenstre ==> punktum
2986:    1  ->  ulrich ==> punktum
2989:    1  ->  understøtte ==> punktum
3007:    1  ->  vagtbureauet ==> punktum
3038:    1  ->  ved ==> punktum
3045:    1  ->  vejret ==> punktum
3059:    1  ->  veste ==> punktum
3060:    1  ->  vesterkÆr ==> punktum
3068:    1  ->  vichy ==> punktum
3073:    1  ->  vidste ==> punktum
3075:    1  ->  vientiane ==> punktum
3085:    1  ->  ville ==> punktum
3089:    1  ->  villet ==> punktum
3100:    1  ->  virginia ==> punktum
3132:    1  ->  vÆret ==> punktum
3138:    1  ->  vÆth ==> punktum
3142:    1  ->  weinreich ==> punktum
3147:    1  ->  west ==> punktum
3149:    1  ->  willemstad ==> punktum
```

```
3150:    1  ->  wimbledon ==> punktum
3153:    1  ->  yamagata ==> punktum
3157:    1  ->  yildiz ==> punktum
3169:    1  ->  Åbnet ==> punktum
3172:    1  ->  Åker ==> punktum
3186:    1  ->  Ægypten ==> punktum
3194:    1  ->  ØbakkegÅrd ==> punktum
3197:    1  ->  ØhlenschlÆger ==> punktum
3205:    1  ->  Østerskoven ==> punktum
3206:    1  ->  ØsterÅgÅrd ==> punktum
```

#### B.5.3.2 Insertions

```
id: (46-r6110007-138)
Scores: (#C #S #D #I) 9 0 0 1
REF:  ******* nul to en tre nitten fem hundrede og halvtreds
HYP:  PUNKTUM nul to en tre nitten fem hundrede og halvtreds
Eval: I
--
id: (46-r6110007-179)
Scores: (#C #S #D #I) 0 2 0 1
REF:  **** RENÉ  TROLLE
HYP:  RENE TROLDE PUNKTUM
Eval: I    S      S
--
id: (46-r6110007-180)
Scores: (#C #S #D #I) 1 1 0 1
REF:  PRIMA    havnen *******
HYP:  PRIMÆRE havnen PUNKTUM
Eval: S               I
--
id: (46-r6110007-181)
Scores: (#C #S #D #I) 0 2 0 4
REF:  **** *** ** ***** ALBERT HALKJÆR
HYP:  ALLE PÅ ET HALVT GIVER  PUNKTUM
Eval: I    I   I  I     S      S
--
id: (46-r6110007-198)
Scores: (#C #S #D #I) 0 1 0 2
REF:  ***** **** FASTEMOSE
HYP:  FASTE MUSE PUNKTUM
Eval: I     I    S
--
id: (46-r6110007-200)
```

```
Scores: (#C #S #D #I) 0 1 0 1
REF:  ********** GÅRSDAL
HYP:  GÅRSDAGENS PUNKTUM
Eval: I           S
--
id: (46-r6110007-201)
Scores: (#C #S #D #I) 0 1 0 2
REF:  **** ***** STADILBY
HYP:  STAT VEDBY PUNKTUM
Eval: I    I     S
--
id: (46-r6110007-218)
Scores: (#C #S #D #I) 2 1 0 1
REF:  de de *** DE
HYP:  de de DET PUNKTUM
Eval:       I   S
--
id: (46-r6110007-220)
Scores: (#C #S #D #I) 2 1 0 1
REF:  kØ kØ **** KØ
HYP:  kØ kØ KØB PUNKTUM
Eval:         I    S
--
id: (46-r6110007-221)
Scores: (#C #S #D #I) 3 0 0 3
REF:  dag * dag * dag *******
HYP:  dag I dag I dag PUNKTUM
Eval:     I     I     I
--
id: (46-r6110007-222)
Scores: (#C #S #D #I) 3 0 0 1
REF:  tog tog tog *******
HYP:  tog tog tog PUNKTUM
```

```
Eval:             I
--
id: (46-r6110007-23)
Scores: (#C #S #D #I) 3 0 0 2
REF:  der der ** der *******
HYP:  der der ER der PUNKTUM
Eval:         I      I
--
id: (46-r6110007-233)
Scores: (#C #S #D #I) 20 0 0 1
REF-1:  de bragte blot bomberne til eksplosion i havet hvor trykket
HYP-1:  de bragte blot bomberne til eksplosion i havet hvor trykket
Eval:
REF-2:  dræbte fiskene i hobetal så de var til at samle *******
HYP-2:  dræbte fiskene i hobetal så de var til at samle PUNKTUM
Eval:                                                    I
--
id: (46-r6110007-235)
Scores: (#C #S #D #I) 4 3 0 1
REF:  det skrev JEG OGSÅ ned i *** NATURLIGE
HYP:  det skrev I   OS    ned i NAT PUNKTUM
Eval:           S   S           I   S
--
id: (46-r6110007-238)
Scores: (#C #S #D #I) 0 3 0 6
REF-1:  ***** ***** **** ********* ******** *********
HYP-1:  TEGNE FILMS REGI TEGNEFILM TRODSIGE TEGNEFILM
Eval: I     I     I    I         I        I
REF-2:  TEGNEFILMSREGI TEGNEFILMSREGI TEGNEFILMSREGI
HYP-2:  TRES           I               PUNKTUM
Eval: S              S              S
--
id: (46-r6110007-24)
```

```
Scores: (#C #S #D #I) 3 0 0 1
REF:  bÅde bÅde bÅde *******
HYP:  bÅde bÅde bÅde PUNKTUM
Eval:                   I
--
id: (46-r6110007-241)
Scores: (#C #S #D #I) 3 0 0 1
REF:  nitten nitten nitten *******
HYP:  nitten nitten nitten PUNKTUM
Eval:                      I
--
id: (46-r6110007-242)
Scores: (#C #S #D #I) 3 0 0 1
REF:  fleste fleste fleste *******
HYP:  fleste fleste fleste PUNKTUM
Eval:                      I
--
id: (46-r6110007-25)
Scores: (#C #S #D #I) 3 0 0 1
REF:  mÅl mÅl mÅl *******
HYP:  mÅl mÅl mÅl PUNKTUM
Eval:                I
--
id: (46-r6110007-250)
Scores: (#C #S #D #I) 6 0 1 1
REF:  kan man leve af ET lille galleri *******
HYP:  kan man leve af ** lille galleri PUNKTUM
Eval:                  D               I
--
id: (46-r6110007-26)
Scores: (#C #S #D #I) 3 0 0 1
REF:  syn syn syn *******
HYP:  syn syn syn PUNKTUM
```

**TITLER I PH.D.SERIEN:**

**2004**

1. Martin Grieger
*Internet-based Electronic Marketplaces and Supply Chain Management*

2. Thomas Basbøll
*LIKENESS*
*A Philosophical Investigation*

3. Morten Knudsen
*Beslutningens vaklen*
*En systemteoretisk analyse of moderniseringen af et amtskommunalt sundhedsvæsen 1980-2000*

4. Lars Bo Jeppesen
*Organizing Consumer Innovation*
*A product development strategy that is based on online communities and allows some firms to benefit from a distributed process of innovation by consumers*

5. Barbara Dragsted
*SEGMENTATION IN TRANSLATION AND TRANSLATION MEMORY SYSTEMS*
*An empirical investigation of cognitive segmentation and effects of integrating a TM system into the translation process*

6. Jeanet Hardis
*Sociale partnerskaber*
*Et socialkonstruktivistisk casestudie af partnerskabsaktørers virkelighedsopfattelse mellem identitet og legitimitet*

7. Henriette Hallberg Thygesen
*System Dynamics in Action*

8. Carsten Mejer Plath
*Strategisk Økonomistyring*

9. Annemette Kjærgaard
*Knowledge Management as Internal Corporate Venturing*
*– a Field Study of the Rise and Fall of a Bottom-Up Process*

10. Knut Arne Hovdal
*De profesjonelle i endring*
Norsk ph.d., ej til salg gennem Samfundslitteratur

11. Søren Jeppesen
*Environmental Practices and Greening Strategies in Small Manufacturing Enterprises in South Africa*
*– A Critical Realist Approach*

12. Lars Frode Frederiksen
*Industriel forskningsledelse*
*– på sporet af mønstre og samarbejde i danske forskningsintensive virksomheder*

13. Martin Jes Iversen
*The Governance of GN Great Nordic*
*– in an age of strategic and structural transitions 1939-1988*

14. Lars Pynt Andersen
*The Rhetorical Strategies of Danish TV Advertising*
*A study of the first fifteen years with special emphasis on genre and irony*

15. Jakob Rasmussen
*Business Perspectives on E-learning*

16. Sof Thrane
*The Social and Economic Dynamics of Networks*
*– a Weberian Analysis of Three Formalised Horizontal Networks*

17. Lene Nielsen
*Engaging Personas and Narrative Scenarios – a study on how a user-centered approach influenced the perception of the design process in the e-business group at AstraZeneca*

18. S.J Valstad
*Organisationsidentitet*
Norsk ph.d., ej til salg gennem Samfundslitteratur

19. Thomas Lyse Hansen
*Six Essays on Pricing and Weather risk in Energy Markets*

20. Sabine Madsen
*Emerging Methods – An Interpretive Study of ISD Methods in Practice*

21. Evis Sinani
*The Impact of Foreign Direct Investment on Efficiency, Productivity Growth and Trade: An Empirical Investigation*

22. Bent Meier Sørensen
*Making Events Work Or, How to Multiply Your Crisis*

23. Pernille Schnoor
*Brand Ethos*
*Om troværdige brand- og virksomhedsidentiteter i et retorisk og diskursteoretisk perspektiv*

24. Sidsel Fabech
*Von welchem Österreich ist hier die Rede?*
*Diskursive forhandlinger og magtkampe mellem rivaliserende nationale identitetskonstruktioner i østrigske pressediskurser*

25. Klavs Odgaard Christensen
*Sprogpolitik og identitetsdannelse i flersprogede forbundsstater*
*Et komparativt studie af Schweiz og Canada*

26. Dana B. Minbaeva
*Human Resource Practices and Knowledge Transfer in Multinational Corporations*

27. Holger Højlund
*Markedets politiske fornuft*
*Et studie af velfærdens organisering i perioden 1990-2003*

28. Christine Mølgaard Frandsen
*A.s erfaring*
*Om mellemværendets praktik i en transformation af mennesket og subjektiviteten*

29. Sine Nørholm Just
*The Constitution of Meaning – A Meaningful Constitution?*
*Legitimacy, identity, and public opinion in the debate on the future of Europe*

**2005**

1. Claus J. Varnes
*Managing product innovation through rules – The role of formal and structured methods in product development*

2. Helle Hedegaard Hein
*Mellem konflikt og konsensus – Dialogudvikling på hospitalsklinikker*

3. Axel Rosenø
*Customer Value Driven Product Innovation – A Study of Market Learning in New Product Development*

4. Søren Buhl Pedersen
*Making space*
*An outline of place branding*

5. Camilla Funck Ellehave
*Differences that Matter*
*An analysis of practices of gender and organizing in contemporary workplaces*

6. Rigmor Madeleine Lond
*Styring af kommunale forvaltninger*

7. Mette Aagaard Andreassen
*Supply Chain versus Supply Chain Benchmarking as a Means to Managing Supply Chains*

8. Caroline Aggestam-Pontoppidan
*From an idea to a standard*
*The UN and the global governance of accountants' competence*

9. Norsk ph.d.

10. Vivienne Heng Ker-ni
*An Experimental Field Study on the*

*Effectiveness of Grocer Media Advertising*
*Measuring Ad Recall and Recognition, Purchase Intentions and Short-Term Sales*

11. Allan Mortensen
*Essays on the Pricing of Corporate Bonds and Credit Derivatives*

12. Remo Stefano Chiari
*Figure che fanno conoscere*
*Itinerario sull'idea del valore cognitivo e espressivo della metafora e di altri tropi da Aristotele e da Vico fino al cognitivismo contemporaneo*

13. Anders Mcllquham-Schmidt
*Strategic Planning and Corporate Performance*
*An integrative research review and a meta-analysis of the strategic planning and corporate performance literature from 1956 to 2003*

14. Jens Geersbro
*The TDF – PMI Case*
*Making Sense of the Dynamics of Business Relationships and Networks*

15 Mette Andersen
*Corporate Social Responsibility in Global Supply Chains*
*Understanding the uniqueness of firm behaviour*

16. Eva Boxenbaum
*Institutional Genesis: Micro – Dynamic Foundations of Institutional Change*

17. Peter Lund-Thomsen
*Capacity Development, Environmental Justice NGOs, and Governance: The Case of South Africa*

18. Signe Jarlov
*Konstruktioner af offentlig ledelse*

19. Lars Stæhr Jensen
*Vocabulary Knowledge and Listening Comprehension in English as a Foreign Language*
*An empirical study employing data elicited from Danish EFL learners*

20. Christian Nielsen
*Essays on Business Reporting*
*Production and consumption of strategic information in the market for information*

21. Marianne Thejls Fischer
*Egos and Ethics of Management Consultants*

22. Annie Bekke Kjær
*Performance management i Proces-innovation*
*– belyst i et social-konstruktivistisk perspektiv*

23. Suzanne Dee Pedersen
*GENTAGELSENS METAMORFOSE*
*Om organisering af den kreative gøren i den kunstneriske arbejdspraksis*

24. Benedikte Dorte Rosenbrink
*Revenue Management*
*Økonomiske, konkurrencemæssige & organisatoriske konsekvenser*

25. Thomas Riise Johansen
*Written Accounts and Verbal Accounts*
*The Danish Case of Accounting and Accountability to Employees*

26. Ann Fogelgren-Pedersen
*The Mobile Internet: Pioneering Users' Adoption Decisions*

27. Birgitte Rasmussen
*Ledelse i fællesskab – de tillidsvalgtes fornyende rolle*

28. Gitte Thit Nielsen
*Remerger*
*– skabende ledelseskræfter i fusion og opkøb*

29. Carmine Gioia
*A MICROECONOMETRIC ANALYSIS OF MERGERS AND ACQUISITIONS*

30. Ole Hinz
*Den effektive forandringsleder: pilot, pædagog eller politiker?*
*Et studie i arbejdslederes meningstilskrivninger i forbindelse med vellykket gennemførelse af ledelsesinitierede forandringsprojekter*

31. Kjell-Åge Gotvassli
*Et praksisbasert perspektiv på dynamiske læringsnettverk i toppidretten*
Norsk ph.d., ej til salg gennem Samfundslitteratur

32. Henriette Langstrup Nielsen
*Linking Healthcare*
*An inquiry into the changing performances of web-based technology for asthma monitoring*

33. Karin Tweddell Levinsen
*Virtuel Uddannelsespraksis*
*Master i IKT og Læring – et casestudie i hvordan proaktiv proceshåndtering kan forbedre praksis i virtuelle læringsmiljøer*

34. Anika Liversage
*Finding a Path*
*Labour Market Life Stories of Immigrant Professionals*

35. Kasper Elmquist Jørgensen
*Studier i samspillet mellem stat og erhvervsliv i Danmark under 1. verdenskrig*

36. Finn Janning
*A DIFFERENT STORY*
*Seduction, Conquest and Discovery*

37. Patricia Ann Plackett
*Strategic Management of the Radical Innovation Process*
*Leveraging Social Capital for Market Uncertainty Management*

**2006**

1. Christian Vintergaard
*Early Phases of Corporate Venturing*

2. Niels Rom-Poulsen
*Essays in Computational Finance*

3. Tina Brandt Husman
*Organisational Capabilities, Competitive Advantage & Project-Based Organisations*
*The Case of Advertising and Creative Good Production*

4. Mette Rosenkrands Johansen
*Practice at the top*
*– how top managers mobilise and use non-financial performance measures*

5. Eva Parum
Corporate governance som strategisk kommunikations- og ledelsesværktøj

6. Susan Aagaard Petersen
*Culture's Influence on Performance Management: The Case of a Danish Company in China*

7. Thomas Nicolai Pedersen
*The Discursive Constitution of Organizational Governance – Between unity and differentiation*
*The Case of the governance of environmental risks by World Bank environmental staff*

8. Cynthia Selin
*Volatile Visions: Transactons in Anticipatory Knowledge*

9. Jesper Banghøj
*Financial Accounting Information and Compensation in Danish Companies*

10. Mikkel Lucas Overby
*Strategic Alliances in Emerging High-Tech Markets: What's the Difference and does it Matter?*

11. Tine Aage
*External Information Acquisition of Industrial Districts and the Impact of Different Knowledge Creation Dimensions*

*A case study of the Fashion and Design Branch of the Industrial District of Montebelluna, NE Italy*

12. Mikkel Flyverbom
*Making the Global Information Society Governable*
*On the Governmentality of Multi-Stakeholder Networks*

13. Anette Grønning
*Personen bag*
*Tilstedeværelse i e-mail som interaktionsform mellem kunde og medarbejder i dansk forsikringskontekst*

14. Jørn Helder
*One Company – One Language?*
*The NN-case*

15. Lars Bjerregaard Mikkelsen
*Differing perceptions of customer value*
*Development and application of a tool for mapping perceptions of customer value at both ends of customer-supplier dyads in industrial markets*

16. Lise Granerud
*Exploring Learning*
*Technological learning within small manufacturers in South Africa*

17. Esben Rahbek Pedersen
*Between Hopes and Realities: Reflections on the Promises and Practices of Corporate Social Responsibility (CSR)*

18. Ramona Samson
*The Cultural Integration Model and European Transformation.*
*The Case of Romania*

**2007**

1. Jakob Vestergaard
*Discipline in The Global Economy*
*Panopticism and the Post-Washington Consensus*

2. Heidi Lund Hansen
*Spaces for learning and working*
*A qualitative study of change of work, management, vehicles of power and social practices in open offices*

3. Sudhanshu Rai
*Exploring the internal dynamics of software development teams during user analysis*
*A tension enabled Institutionalization Model; "Where process becomes the objective"*

4. Norsk ph.d.
Ej til salg gennem Samfundslitteratur

5. Serden Ozcan
*EXPLORING HETEROGENEITY IN ORGANIZATIONAL ACTIONS AND OUTCOMES*
*A Behavioural Perspective*

6. Kim Sundtoft Hald
*Inter-organizational Performance Measurement and Management in Action*
*– An Ethnography on the Construction of Management, Identity and Relationships*

7. Tobias Lindeberg
*Evaluative Technologies*
*Quality and the Multiplicity of Performance*

8. Merete Wedell-Wedellsborg
*Den globale soldat*
*Identitetsdannelse og identitetsledelse i multinationale militære organisationer*

9. Lars Frederiksen
*Open Innovation Business Models*
*Innovation in firm-hosted online user communities and inter-firm project ventures in the music industry*
*– A collection of essays*

10. Jonas Gabrielsen
*Retorisk toposlære – fra statisk 'sted' til persuasiv aktivitet*

11. Christian Moldt-Jørgensen
*Fra meningsløs til meningsfuld evaluering.*
*Anvendelsen af studentertilfredshedsmålinger på de korte og mellemlange videregående uddannelser set fra et psykodynamisk systemperspektiv*

12. Ping Gao
*Extending the application of actor-network theory*
*Cases of innovation in the telecommunications industry*

13. Peter Mejlby
*Frihed og fængsel, en del af den samme drøm?*
*Et phronetisk baseret casestudie af frigørelsens og kontrollens sameksistens i værdibaseret ledelse!*

14. Kristina Birch
*Statistical Modelling in Marketing*

15. Signe Poulsen
*Sense and sensibility:*
*The language of emotional appeals in insurance marketing*

16. Anders Bjerre Trolle
*Essays on derivatives pricing and dynamic asset allocation*

17. Peter Feldhütter
*Empirical Studies of Bond and Credit Markets*

18. Jens Henrik Eggert Christensen
*Default and Recovery Risk Modeling and Estimation*

19. Maria Theresa Larsen
*Academic Enterprise: A New Mission for Universities or a Contradiction in Terms?*
*Four papers on the long-term implications of increasing industry involvement and commercialization in academia*

20. Morten Wellendorf
*Postimplementering af teknologi i den offentlige forvaltning*
*Analyser af en organisations kontinuerlige arbejde med informationsteknologi*

21. Ekaterina Mhaanna
*Concept Relations for Terminological Process Analysis*

22. Stefan Ring Thorbjørnsen
*Forsvaret i forandring*
*Et studie i officerers kapabiliteter under påvirkning af omverdenens forandringspres mod øget styring og læring*

23. Christa Breum Amhøj
*Det selvskabte medlemskab om managementstaten, dens styringsteknologier og indbyggere*

24. Karoline Bromose
*Between Technological Turbulence and Operational Stability*
*– An empirical case study of corporate venturing in TDC*

25. Susanne Justesen
*Navigating the Paradoxes of Diversity in Innovation Practice*
*– A Longitudinal study of six very different innovation processes – in practice*

26. Luise Noring Henler
*Conceptualising successful supply chain partnerships*
*– Viewing supply chain partnerships from an organisational culture perspective*

27. Mark Mau
*Kampen om telefonen*
*Det danske telefonvæsen under den tyske besættelse 1940-45*

28. Jakob Halskov
*The semiautomatic expansion of existing terminological ontologies using knowledge patterns discovered*

*on the WWW – an implementation and evaluation*

29. Gergana Koleva
*European Policy Instruments Beyond Networks and Structure: The Innovative Medicines Initiative*

30. Christian Geisler Asmussen
*Global Strategy and International Diversity: A Double-Edged Sword?*

31. Christina Holm-Petersen
*Stolthed og fordom*
*Kultur- og identitetsarbejde ved skabelsen af en ny sengeafdeling gennem fusion*

32. Hans Peter Olsen
*Hybrid Governance of Standardized States*
*Causes and Contours of the Global Regulation of Government Auditing*

33. Lars Bøge Sørensen
*Risk Management in the Supply Chain*

34. Peter Aagaard
*Det unikkes dynamikker*
*De institutionelle mulighedsbetingelser bag den individuelle udforskning i professionelt og frivilligt arbejde*

35. Yun Mi Antorini
*Brand Community Innovation*
*An Intrinsic Case Study of the Adult Fans of LEGO Community*

36. Joachim Lynggaard Boll
*Labor Related Corporate Social Performance in Denmark*
*Organizational and Institutional Perspectives*

**2008**

1. Frederik Christian Vinten
*Essays on Private Equity*

2. Jesper Clement
*Visual Influence of Packaging Design on In-Store Buying Decisions*

3. Marius Brostrøm Kousgaard
*Tid til kvalitetsmåling?*
*– Studier af indrulleringsprocesser i forbindelse med introduktionen af kliniske kvalitetsdatabaser i speciallægepraksissektoren*

4. Irene Skovgaard Smith
*Management Consulting in Action*
*Value creation and ambiguity in client-consultant relations*

5. Anders Rom
*Management accounting and integrated information systems*
*How to exploit the potential for management accounting of information technology*

6. Marina Candi
*Aesthetic Design as an Element of Service Innovation in New Technology-based Firms*

7. Morten Schnack
*Teknologi og tværfaglighed*
*– en analyse af diskussionen omkring indførelse af EPJ på en hospitalsafdeling*

8. Helene Balslev Clausen
*Juntos pero no revueltos – un estudio sobre emigrantes norteamericanos en un pueblo mexicano*

9. Lise Justesen
*Kunsten at skrive revisionsrapporter. En beretning om forvaltningsrevisionens beretninger*

10. Michael E. Hansen
*The politics of corporate responsibility: CSR and the governance of child labor and core labor rights in the 1990s*

11. Anne Roepstorff
*Holdning for handling – en etnologisk undersøgelse af Virksomheders Sociale Ansvar/CSR*

12. Claus Bajlum
*Essays on Credit Risk and Credit Derivatives*

13. Anders Bojesen
*The Performative Power of Competence – an Inquiry into Subjectivity and Social Technologies at Work*

14. Satu Reijonen
*Green and Fragile*
*A Study on Markets and the Natural Environment*

15. Ilduara Busta
*Corporate Governance in Banking*
*A European Study*

16. Kristian Anders Hvass
*A Boolean Analysis Predicting Industry Change: Innovation, Imitation & Business Models*
*The Winning Hybrid: A case study of isomorphism in the airline industry*

17. Trine Paludan
*De uvidende og de udviklingsparate*
*Identitet som mulighed og restriktion blandt fabriksarbejdere på det aftayloriserede fabriksgulv*

18. Kristian Jakobsen
*Foreign market entry in transition economies: Entry timing and mode choice*

19. Jakob Elming
*Syntactic reordering in statistical machine translation*

20. Lars Brømsøe Termansen
*Regional Computable General Equilibrium Models for Denmark*
*Three papers laying the foundation for regional CGE models with agglomeration characteristics*

21. Mia Reinholt
*The Motivational Foundations of Knowledge Sharing*

22. Frederikke Krogh-Meibom
*The Co-Evolution of Institutions and Technology*
*– A Neo-Institutional Understanding of Change Processes within the Business Press – the Case Study of Financial Times*

23. Peter D. Ørberg Jensen
*OFFSHORING OF ADVANCED AND HIGH-VALUE TECHNICAL SERVICES: ANTECEDENTS, PROCESS DYNAMICS AND FIRMLEVEL IMPACTS*

24. Pham Thi Song Hanh
*Functional Upgrading, Relational Capability and Export Performance of Vietnamese Wood Furniture Producers*

25. Mads Vangkilde
*Why wait?*
*An Exploration of first-mover advantages among Danish e-grocers through a resource perspective*

26. Hubert Buch-Hansen
*Rethinking the History of European Level Merger Control*
*A Critical Political Economy Perspective*

**2009**

1. Vivian Lindhardsen
*From Independent Ratings to Communal Ratings: A Study of CWA Raters' Decision-Making Behaviours*

2. Guðrið Weihe
*Public-Private Partnerships: Meaning and Practice*

3. Chris Nøkkentved
*Enabling Supply Networks with Collaborative Information Infrastructures*
*An Empirical Investigation of Business Model Innovation in Supplier Relationship Management*

4. Sara Louise Muhr
*Wound, Interrupted – On the Vulnerability of Diversity Management*

5. Christine Sestoft
*Forbrugeradfærd i et Stats- og Livsformsteoretisk perspektiv*

6. Michael Pedersen
*Tune in, Breakdown, and Reboot: On the production of the stress-fit self-managing employee*

7. Salla Lutz
*Position and Reposition in Networks – Exemplified by the Transformation of the Danish Pine Furniture Manufacturers*

8. Jens Forssbæck
*Essays on market discipline in commercial and central banking*

9. Tine Murphy
*Sense from Silence – A Basis for Organised Action*
*How do Sensemaking Processes with Minimal Sharing Relate to the Reproduction of Organised Action?*

10. Sara Malou Strandvad
*Inspirations for a new sociology of art: A sociomaterial study of development processes in the Danish film industry*

11. Nicolaas Mouton
*On the evolution of social scientific metaphors:*
*A cognitive-historical enquiry into the divergent trajectories of the idea that collective entities – states and societies, cities and corporations – are biological organisms.*

12. Lars Andreas Knutsen
*Mobile Data Services:*
*Shaping of user engagements*

13. Nikolaos Theodoros Korfiatis
*Information Exchange and Behavior*
*A Multi-method Inquiry on Online Communities*

14. Jens Albæk
*Forestillinger om kvalitet og tværfaglighed på sygehuse*
*– skabelse af forestillinger i læge- og plejegrupperne angående relevans af nye idéer om kvalitetsudvikling gennem tolkningsprocesser*

15. Maja Lotz
*The Business of Co-Creation – and the Co-Creation of Business*

16. Gitte P. Jakobsen
*Narrative Construction of Leader Identity in a Leader Development Program Context*

17. Dorte Hermansen
*"Living the brand" som en brandorienteret dialogisk praxis:*
*Om udvikling af medarbejdernes brandorienterede dømmekraft*

18. Aseem Kinra
*Supply Chain (logistics) Environmental Complexity*

19. Michael Nørager
How to manage SMEs through the transformation from non innovative to innovative?

20. Kristin Wallevik
*Corporate Governance in Family Firms*
*The Norwegian Maritime Sector*

21. Bo Hansen Hansen
*Beyond the Process*
*Enriching Software Process Improvement with Knowledge Management*

22. Annemette Skot-Hansen
*Franske adjektivisk afledte adverbier, der tager præpositionssyntagmer indledt med præpositionen à som argumenter*
*En valensgrammatisk undersøgelse*

23. Line Gry Knudsen
*Collaborative R&D Capabilities*
*In Search of Micro-Foundations*

24. Christian Scheuer
*Employers meet employees*
*Essays on sorting and globalization*

25. Rasmus Johnsen
*The Great Health of Melancholy*
*A Study of the Pathologies of Performativity*

26. Ha Thi Van Pham
*Internationalization, Competitiveness Enhancement and Export Performance of Emerging Market Firms:*
*Evidence from Vietnam*

27. Henriette Balieu
*Kontrolbegrebets betydning for kausativalternationen i spansk*
*En kognitiv-typologisk analyse*

**2010**

1. Yen Tran
*Organizing Innovationin Turbulent Fashion Market*
*Four papers on how fashion firms create and appropriate innovation value*

2. Anders Raastrup Kristensen
*Metaphysical Labour*
*Flexibility, Performance and Commitment in Work-Life Management*

3. Margrét Sigrún Sigurdardottir
*Dependently independent*
*Co-existence of institutional logics in the recorded music industry*

4. Ásta Dis Óladóttir
*Internationalization from a small domestic base:*
*An empirical analysis of Economics and Management*

5. Christine Secher
*E-deltagelse i praksis – politikernes og forvaltningens medkonstruktion og konsekvenserne heraf*

6. Marianne Stang Våland
*What we talk about when we talk about space:*
*End User Participation between Processes of Organizational and Architectural Design*

7. Rex Degnegaard
*Strategic Change Management*
*Change Management Challenges in the Danish Police Reform*

8. Ulrik Schultz Brix
*Værdi i rekruttering – den sikre beslutning*
*En pragmatisk analyse af perception og synliggørelse af værdi i rekrutterings- og udvælgelsesarbejdet*

9. Jan Ole Similä
*Kontraktsledelse*
*Relasjonen mellom virksomhetsledelse og kontraktshåndtering, belyst via fire norske virksomheter*

10. Susanne Boch Waldorff
*Emerging Organizations: In between local translation, institutional logics and discourse*

11. Brian Kane
*Performance Talk*
*Next Generation Management of Organizational Performance*

12. Lars Ohnemus
*Brand Thrust: Strategic Branding and Shareholder Value*
*An Empirical Reconciliation of two Critical Concepts*

13. Jesper Schlamovitz
*Håndtering af usikkerhed i film- og byggeprojekter*

14. Tommy Moesby-Jensen
*Det faktiske livs forbindtlighed*
*Førsokratisk informeret, ny-aristotelisk ἦθος-tænkning hos Martin Heidegger*

15. Christian Fich
*Two Nations Divided by Common Values*
*French National Habitus and the Rejection of American Power*

16. Peter Beyer
*Processer, sammenhængskraft og fleksibilitet*
*Et empirisk casestudie af omstillingsforløb i fire virksomheder*

17. Adam Buchhorn
*Markets of Good Intentions*
*Constructing and Organizing Biogas Markets Amid Fragility and Controversy*

18. Cecilie K. Moesby-Jensen
*Social læring og fælles praksis*
*Et mixed method studie, der belyser læringskonsekvenser af et lederkursus for et praksisfællesskab af offentlige mellemledere*

19. Heidi Boye
*Fødevarer og sundhed i senmodernismen*
*– En indsigt i hyggefænomenet og de relaterede fødevarepraksisser*

20. Kristine Munkgård Pedersen
*Flygtige forbindelser og midlertidige mobiliseringer*
*Om kulturel produktion på Roskilde Festival*

21. Oliver Jacob Weber
*Causes of Intercompany Harmony in Business Markets – An Empirical Investigation from a Dyad Perspective*

22. Susanne Ekman
*Authority and Autonomy*
*Paradoxes of Modern Knowledge Work*

23. Anette Frey Larsen
*Kvalitetsledelse på danske hospitaler*
*– Ledelsernes indflydelse på introduktion og vedligeholdelse af kvalitetsstrategier i det danske sundhedsvæsen*

24. Toyoko Sato
*Performativity and Discourse: Japanese Advertisements on the Aesthetic Education of Desire*

25. Kenneth Brinch Jensen
*Identifying the Last Planner System*
*Lean management in the construction industry*

26. Javier Busquets
*Orchestrating Network Behavior for Innovation*

27. Luke Patey
*The Power of Resistance: India's National Oil Company and International Activism in Sudan*

28. Mette Vedel
*Value Creation in Triadic Business Relationships. Interaction, Interconnection and Position*

29. Kristian Tørning
*Knowledge Management Systems in Practice – A Work Place Study*

30. Qingxin Shi
*An Empirical Study of Thinking Aloud Usability Testing from a Cultural Perspective*

31. Tanja Juul Christiansen
*Corporate blogging: Medarbejderes kommunikative handlekraft*

32. Malgorzata Ciesielska
*Hybrid Organisations.*
*A study of the Open Source – business setting*

33. Jens Dick-Nielsen
*Three Essays on Corporate Bond Market Liquidity*

34. Sabrina Speiermann
*Modstandens Politik*
*Kampagnestyring i Velfærdsstaten.*
*En diskussion af trafikkampagners styringspotentiale*

35. Julie Uldam
*Fickle Commitment. Fostering political engagement in 'the flighty world of online activism'*

36. Annegrete Juul Nielsen
*Traveling technologies and transformations in health care*

37. Athur Mühlen-Schulte
*Organising Development*
*Power and Organisational Reform in the United Nations Development Programme*

38. Louise Rygaard Jonas
*Branding på butiksgulvet*
*Et case-studie af kultur- og identitetsarbejdet i Kvickly*

**2011**

1. Stefan Fraenkel
*Key Success Factors for Sales Force Readiness during New Product Launch*
*A Study of Product Launches in the Swedish Pharmaceutical Industry*

2. Christian Plesner Rossing
*International Transfer Pricing in Theory and Practice*

3. Tobias Dam Hede
*Samtalekunst og ledelsesdisciplin*
*– en analyse af coachingsdiskursens genealogi og governmentality*

4. Kim Pettersson
*Essays on Audit Quality, Auditor Choice, and Equity Valuation*

5. Henrik Merkelsen
*The expert-lay controversy in risk research and management. Effects of institutional distances. Studies of risk definitions, perceptions, management and communication*

6. Simon S. Torp
*Employee Stock Ownership:*
*Effect on Strategic Management and Performance*

7. Mie Harder
*Internal Antecedents of Management Innovation*

8. Ole Helby Petersen
*Public-Private Partnerships: Policy and Regulation – With Comparative and Multi-level Case Studies from Denmark and Ireland*

9. Morten Krogh Petersen
*'Good' Outcomes. Handling Multiplicity in Government Communication*

10. Kristian Tangsgaard Hvelplund
*Allocation of cognitive resources in translation - an eye-tracking and key-logging study*

11. Moshe Yonatany
*The Internationalization Process of Digital Service Providers*

12. Anne Vestergaard
*Distance and Suffering*
*Humanitarian Discourse in the age of Mediatization*

13. Thorsten Mikkelsen
*Personligsheds indflydelse på forretningsrelationer*

14. Jane Thostrup Jagd
*Hvorfor fortsætter fusionsbølgen udover "the tipping point"?*
*– en empirisk analyse af information og kognitioner om fusioner*

15. Gregory Gimpel
*Value-driven Adoption and Consumption of Technology: Understanding Technology Decision Making*

16. Thomas Stengade Sønderskov
*Den nye mulighed*
*Social innovation i en forretningsmæssig kontekst*

17. Jeppe Christoffersen
*Donor supported strategic alliances in developing countries*

18. Vibeke Vad Baunsgaard
*Dominant Ideological Modes of Rationality: Cross functional*

*integration in the process of product innovation*

19. Throstur Olaf Sigurjonsson
*Governance Failure and Icelands's Financial Collapse*

20. Allan Sall Tang Andersen
*Essays on the modeling of risks in interest-rate and inflation markets*

21. Heidi Tscherning
*Mobile Devices in Social Contexts*

22. Birgitte Gorm Hansen
*Adapting in the Knowledge Economy Lateral Strategies for Scientists and Those Who Study Them*

23. Kristina Vaarst Andersen
*Optimal Levels of Embeddedness The Contingent Value of Networked Collaboration*

24. Justine Grønbæk Pors
*Noisy Management A History of Danish School Governing from 1970-2010*

25. Stefan Linder
*Micro-foundations of Strategic Entrepreneurship Essays on Autonomous Strategic Action*

26. Xin Li
*Toward an Integrative Framework of National Competitiveness An application to China*

27. Rune Thorbjørn Clausen
*Værdifuld arkitektur Et eksplorativt studie af bygningers rolle i virksomheders værdiskabelse*

28. Monica Viken
*Markedsundersøkelser som bevis i varemerke- og markedsføringsrett*

29. Christian Wymann
*Tattooing The Economic and Artistic Constitution of a Social Phenomenon*

30. Sanne Frandsen
*Productive Incoherence A Case Study of Branding and Identity Struggles in a Low-Prestige Organization*

31. Mads Stenbo Nielsen
*Essays on Correlation Modelling*

32. Ivan Häuser
*Følelse og sprog Etablering af en ekspressiv kategori, eksemplificeret på russisk*

33. Sebastian Schwenen
*Security of Supply in Electricity Markets*

**2012**

1. Peter Holm Andreasen
*The Dynamics of Procurement Management - A Complexity Approach*

2. Martin Haulrich
*Data-Driven Bitext Dependency Parsing and Alignment*

3. Line Kirkegaard
*Konsulenten i den anden nat En undersøgelse af det intense arbejdsliv*

4. Tonny Stenheim
*Decision usefulness of goodwill under IFRS*

5. Morten Lind Larsen
*Produktivitet, vækst og velfærd Industrirådet og efterkrigstidens Danmark 1945 - 1958*

6. Petter Berg
*Cartel Damages and Cost Asymmetries*

7. Lynn Kahle
*Experiential Discourse in Marketing A methodical inquiry into practice and theory*

8. Anne Roelsgaard Obling
*Management of Emotions in Accelerated Medical Relationships*

9. Thomas Frandsen
*Managing Modularity of Service Processes Architecture*

10. Carina Christine Skovmøller
*CSR som noget særligt*
*Et casestudie om styring og menings-skabelse i relation til CSR ud fra en intern optik*

11. Michael Tell
*Fradragsbeskæring af selskabers finansieringsudgifter*
*En skatteretlig analyse af SEL §§ 11, 11B og 11C*

12. Morten Holm
*Customer Profitability Measurement Models*
*Their Merits and Sophistication across Contexts*

13. Katja Joo Dyppel
*Beskatning af derivater*
*En analyse af dansk skatteret*

14. Esben Anton Schultz
*Essays in Labor Economics*
*Evidence from Danish Micro Data*

15. Carina Risvig Hansen
*"Contracts not covered, or not fully covered, by the Public Sector Directive"*

16. Anja Svejgaard Pors
*Iværksættelse af kommunikation*
*- patientfigurer i hospitalets strategiske kommunikation*

17. Frans Bévort
*Making sense of management with logics*
*An ethnographic study of accountants who become managers*

18. René Kallestrup
*The Dynamics of Bank and Sovereign Credit Risk*

19. Brett Crawford
*Revisiting the Phenomenon of Interests in Organizational Institutionalism*
*The Case of U.S. Chambers of Commerce*

20. Mario Daniele Amore
*Essays on Empirical Corporate Finance*

21. Arne Stjernholm Madsen
*The evolution of innovation strategy*
*Studied in the context of medical device activities at the pharmaceutical company Novo Nordisk A/S in the period 1980-2008*

22. Jacob Holm Hansen
*Is Social Integration Necessary for Corporate Branding?*
*A study of corporate branding strategies at Novo Nordisk*

23. Stuart Webber
*Corporate Profit Shifting and the Multinational Enterprise*

24. Helene Ratner
*Promises of Reflexivity*
*Managing and Researching Inclusive Schools*

25. Therese Strand
*The Owners and the Power: Insights from Annual General Meetings*

26. Robert Gavin Strand
*In Praise of Corporate Social Responsibility Bureaucracy*

27. Nina Sormunen
*Auditor's going-concern reporting*
*Reporting decision and content of the report*

28. John Bang Mathiasen
*Learning within a product development working practice:*
*- an understanding anchored in pragmatism*

29. Philip Holst Riis
*Understanding Role-Oriented Enterprise Systems: From Vendors to Customers*

30. Marie Lisa Dacanay
*Social Enterprises and the Poor*
*Enhancing Social Entrepreneurship and Stakeholder Theory*

31. Fumiko Kano Glückstad
*Bridging Remote Cultures: Cross-lingual concept mapping based on the information receiver's prior-knowledge*

32. Henrik Barslund Fosse
*Empirical Essays in International Trade*

33. Peter Alexander Albrecht
*Foundational hybridity and its reproduction*
*Security sector reform in Sierra Leone*

34. Maja Rosenstock
*CSR - hvor svært kan det være? Kulturanalytisk casestudie om udfordringer og dilemmaer med at forankre Coops CSR-strategi*

35. Jeanette Rasmussen
*Tweens, medier og forbrug*
*Et studie af 10-12 årige danske børns brug af internettet, opfattelse og forståelse af markedsføring og forbrug*

36. Ib Tunby Gulbrandsen
*'This page is not intended for a US Audience'*
*A five-act spectacle on online communication, collaboration & organization.*

37. Kasper Aalling Teilmann
*Interactive Approaches to Rural Development*

38. Mette Mogensen
*The Organization(s) of Well-being and Productivity*
*(Re)assembling work in the Danish Post*

39. Søren Friis Møller
*From Disinterestedness to Engagement*
*Towards Relational Leadership In the Cultural Sector*

40. Nico Peter Berhausen
*Management Control, Innovation and Strategic Objectives – Interactions and Convergence in Product Development Networks*

41. Balder Onarheim
*Creativity under Constraints*
*Creativity as Balancing 'Constrainedness'*

42. Haoyong Zhou
*Essays on Family Firms*

43. Elisabeth Naima Mikkelsen
*Making sense of organisational conflict*
*An empirical study of enacted sense-making in everyday conflict at work*

**2013**

1. Jacob Lyngsie
*Entrepreneurship in an Organizational Context*

2. Signe Groth-Brodersen
*Fra ledelse til selvet*
*En socialpsykologisk analyse af forholdet imellem selvledelse, ledelse og stress i det moderne arbejdsliv*

3. Nis Høyrup Christensen
*Shaping Markets: A Neoinstitutional Analysis of the Emerging Organizational Field of Renewable Energy in China*

4. Christian Edelvold Berg
*As a matter of size*
*THE IMPORTANCE OF CRITICAL MASS AND THE CONSEQUENCES OF SCARCITY FOR TELEVISION MARKETS*

5. Christine D. Isakson
*Coworker Influence and Labor Mobility*
*Essays on Turnover, Entrepreneurship and Location Choice in the Danish Maritime Industry*

6. Niels Joseph Jerne Lennon
*Accounting Qualities in Practice*
*Rhizomatic stories of representational faithfulness, decision making and control*

7. Shannon O'Donnell
*Making Ensemble Possible*
*How special groups organize for collaborative creativity in conditions of spatial variability and distance*

8. Robert W. D. Veitch
*Access Decisions in a Partly-Digital World*
*Comparing Digital Piracy and Legal Modes for Film and Music*

9. Marie Mathiesen
*Making Strategy Work*
*An Organizational Ethnography*

10. Arisa Shollo
*The role of business intelligence in organizational decision-making*

11. Mia Kaspersen
*The construction of social and environmental reporting*

12. Marcus Møller Larsen
*The organizational design of offshoring*

13. Mette Ohm Rørdam
*EU Law on Food Naming*
*The prohibition against misleading names in an internal market context*

14. Hans Peter Rasmussen
*GIV EN GED!*
*Kan giver-idealtyper forklare støtte til velgørenhed og understøtte relationsopbygning?*

15. Ruben Schachtenhaufen
*Fonetisk reduktion i dansk*

16. Peter Koerver Schmidt
*Dansk CFC-beskatning*
*I et internationalt og komparativt perspektiv*

17. Morten Froholdt
*Strategi i den offentlige sektor*
*En kortlægning af styringsmæssig kontekst, strategisk tilgang, samt anvendte redskaber og teknologier for udvalgte danske statslige styrelser*

18. Annette Camilla Sjørup
*Cognitive effort in metaphor translation*
*An eye-tracking and key-logging study*

19. Tamara Stucchi
*The Internationalization of Emerging Market Firms:*
*A Context-Specific Study*

20. Thomas Lopdrup-Hjorth
*"Let's Go Outside":*
*The Value of Co-Creation*

21. Ana Alačovska
*Genre and Autonomy in Cultural Production*
*The case of travel guidebook production*

22. Marius Gudmand-Høyer
*Stemningssindssygdommenes historie i det 19. århundrede*
*Omtydningen af melankolien og manien som bipolære stemningslidelser i dansk sammenhæng under hensyn til dannelsen af det moderne følelseslivs relative autonomi.*
*En problematiserings- og erfarings-analytisk undersøgelse*

23. Lichen Alex Yu
*Fabricating an S&OP Process*
*Circulating References and Matters of Concern*

24. Esben Alfort
*The Expression of a Need*
*Understanding search*

25. Trine Pallesen
*Assembling Markets for Wind Power*
*An Inquiry into the Making of Market Devices*

26. Anders Koed Madsen
*Web-Visions*
*Repurposing digital traces to organize social attention*

27. Lærke Højgaard Christiansen
*BREWING ORGANIZATIONAL RESPONSES TO INSTITUTIONAL LOGICS*

28. Tommy Kjær Lassen
*EGENTLIG SELVLEDELSE*
*En ledelsesfilosofisk afhandling om selvledelsens paradoksale dynamik og eksistentielle engagement*

29. Morten Rossing
*Local Adaption and Meaning Creation in Performance Appraisal*

30. Søren Obed Madsen
*Lederen som oversætter*
*Et oversættelsesteoretisk perspektiv på strategisk arbejde*

31. Thomas Høgenhaven
*Open Government Communities*
*Does Design Affect Participation?*

32. Kirstine Zinck Pedersen
*Failsafe Organizing?*
*A Pragmatic Stance on Patient Safety*

33. Anne Petersen
*Hverdagslogikker i psykiatrisk arbejde*
*En institutionsetnografisk undersøgelse af hverdagen i psykiatriske organisationer*

34. Didde Maria Humle
*Fortællinger om arbejde*

35. Mark Holst-Mikkelsen
*Strategieksekvering i praksis*
*– barrierer og muligheder!*

36. Malek Maalouf
*Sustaining lean*
*Strategies for dealing with organizational paradoxes*

37. Nicolaj Tofte Brenneche
*Systemic Innovation In The Making*
*The Social Productivity of Cartographic Crisis and Transitions in the Case of SEEIT*

38. Morten Gylling
*The Structure of Discourse*
*A Corpus-Based Cross-Linguistic Study*

39. Binzhang YANG
*Urban Green Spaces for Quality Life*
*- Case Study: the landscape architecture for people in Copenhagen*

40. Michael Friis Pedersen
*Finance and Organization:*
*The Implications for Whole Farm Risk Management*

41. Even Fallan
*Issues on supply and demand for environmental accounting information*

42. Ather Nawaz
*Website user experience*
*A cross-cultural study of the relation between users´ cognitive style, context of use, and information architecture of local websites*

43. Karin Beukel
*The Determinants for Creating Valuable Inventions*

44. Arjan Markus
*External Knowledge Sourcing and Firm Innovation*
*Essays on the Micro-Foundations of Firms' Search for Innovation*

**2014**

1. Solon Moreira
*Four Essays on Technology Licensing and Firm Innovation*

2. Karin Strzeletz Ivertsen
*Partnership Drift in Innovation Processes*
*A study of the Think City electric car development*

3. Kathrine Hoffmann Pii
*Responsibility Flows in Patient-centred Prevention*

4. Jane Bjørn Vedel
*Managing Strategic Research*
*An empirical analysis of science-industry collaboration in a pharmaceutical company*

5. Martin Gylling
*Processuel strategi i organisationer*
*Monografi om dobbeltheden i tænkning af strategi, dels som vidensfelt i organisationsteori, dels som kunstnerisk tilgang til at skabe i erhvervsmæssig innovation*

6. Linne Marie Lauesen
*Corporate Social Responsibility in the Water Sector: How Material Practices and their Symbolic and Physical Meanings Form a Colonising Logic*

7. Maggie Qiuzhu Mei
*LEARNING TO INNOVATE: The role of ambidexterity, standard, and decision process*

8. Inger Høedt-Rasmussen
*Developing Identity for Lawyers Towards Sustainable Lawyering*

9. Sebastian Fux
*Essays on Return Predictability and Term Structure Modelling*

10. Thorbjørn N. M. Lund-Poulsen
*Essays on Value Based Management*

11. Oana Brindusa Albu
*Transparency in Organizing: A Performative Approach*

12. Lena Olaison
*Entrepreneurship at the limits*

13. Hanne Sørum
*DRESSED FOR WEB SUCCESS? An Empirical Study of Website Quality in the Public Sector*

14. Lasse Folke Henriksen
*Knowing networks How experts shape transnational governance*

15. Maria Halbinger
*Entrepreneurial Individuals Empirical Investigations into Entrepreneurial Activities of Hackers and Makers*

16. Robert Spliid
*Kapitalfondenes metoder og kompetencer*

17. Christiane Stelling
*Public-private partnerships & the need, development and management of trusting A processual and embedded exploration*

18. Marta Gasparin
*Management of design as a translation process*

19. Kåre Moberg
*Assessing the Impact of Entrepreneurship Education From ABC to PhD*

20. Alexander Cole
*Distant neighbors Collective learning beyond the cluster*

21. Martin Møller Boje Rasmussen
*Is Competitiveness a Question of Being Alike? How the United Kingdom, Germany and Denmark Came to Compete through their Knowledge Regimes from 1993 to 2007*

22. Anders Ravn Sørensen
*Studies in central bank legitimacy, currency and national identity Four cases from Danish monetary history*

23. Nina Bellak
*Can Language be Managed in International Business? Insights into Language Choice from a Case Study of Danish and Austrian Multinational Corporations (MNCs)*

24. Rikke Kristine Nielsen
*Global Mindset as Managerial Meta-competence and Organizational Capability: Boundary-crossing Leadership Cooperation in the MNC The Case of 'Group Mindset' in Solar A/S.*

25. Rasmus Koss Hartmann
*User Innovation inside government Towards a critically performative foundation for inquiry*

26. Kristian Gylling Olesen
*Flertydig og emergerende ledelse i folkeskolen*
*Et aktør-netværksteoretisk ledelses-studie af politiske evalueringsreformers betydning for ledelse i den danske folkeskole*

27. Troels Riis Larsen
*Kampen om Danmarks omdømme 1945-2010*
*Omdømmearbejde og omdømmepolitik*

28. Klaus Majgaard
*Jagten på autenticitet i offentlig styring*

29. Ming Hua Li
*Institutional Transition and Organizational Diversity: Differentiated internationalization strategies of emerging market state-owned enterprises*

30. Sofie Blinkenberg Federspiel
*IT, organisation og digitalisering: Institutionelt arbejde i den kommunale digitaliseringsproces*

31. Elvi Weinreich
*Hvilke offentlige ledere er der brug for når velfærdstænkningen flytter sig – er Diplomuddannelsens lederprofil svaret?*

32. Ellen Mølgaard Korsager
*Self-conception and image of context in the growth of the firm – A Penrosian History of Fiberline Composites*

33. Else Skjold
*The Daily Selection*

34. Marie Louise Conradsen
*The Cancer Centre That Never Was The Organisation of Danish Cancer Research 1949-1992*

35. Virgilio Failla
*Three Essays on the Dynamics of Entrepreneurs in the Labor Market*

36. Nicky Nedergaard
*Brand-Based Innovation*
*Relational Perspectives on Brand Logics and Design Innovation Strategies and Implementation*

37. Mads Gjedsted Nielsen
*Essays in Real Estate Finance*

38. Kristin Martina Brandl
*Process Perspectives on Service Offshoring*

39. Mia Rosa Koss Hartmann
*In the gray zone*
*With police in making space for creativity*

40. Karen Ingerslev
*Healthcare Innovation under The Microscope*
*Framing Boundaries of Wicked Problems*

41. Tim Neerup Themsen
*Risk Management in large Danish public capital investment programmes*

**2015**

1. Jakob Ion Wille
*Film som design*
*Design af levende billeder i film og tv-serier*

2. Christiane Mossin
*Interzones of Law and Metaphysics*
*Hierarchies, Logics and Foundations of Social Order seen through the Prism of EU Social Rights*

3. Thomas Tøth
*TRUSTWORTHINESS: ENABLING GLOBAL COLLABORATION*
*An Ethnographic Study of Trust, Distance, Control, Culture and Boundary Spanning within Offshore Outsourcing of IT Services*

4. Steven Højlund
*Evaluation Use in Evaluation Systems – The Case of the European Commission*

5. Julia Kirch Kirkegaard
*AMBIGUOUS WINDS OF CHANGE – OR FIGHTING AGAINST WINDMILLS IN CHINESE WIND POWER*
*A CONSTRUCTIVIST INQUIRY INTO CHINA'S PRAGMATICS OF GREEN MARKETISATION MAPPING CONTROVERSIES OVER A POTENTIAL TURN TO QUALITY IN CHINESE WIND POWER*

6. Michelle Carol Antero
*A Multi-case Analysis of the Development of Enterprise Resource Planning Systems (ERP) Business Practices*

Morten Friis-Olivarius
*The Associative Nature of Creativity*

7. Mathew Abraham
*New Cooperativism:*
*A study of emerging producer organisations in India*

8. Stine Hedegaard
*Sustainability-Focused Identity: Identity work performed to manage, negotiate and resolve barriers and tensions that arise in the process of constructing or ganizational identity in a sustainability context*

9. Cecilie Glerup
*Organizing Science in Society – the conduct and justification of resposible research*

10. Allan Salling Pedersen
*Implementering af ITIL® IT-governance - når best practice konflikter med kulturen Løsning af implementerings-problemer gennem anvendelse af kendte CSF i et aktionsforskningsforløb.*

11. Nihat Misir
*A Real Options Approach to Determining Power Prices*

12. Mamdouh Medhat
*MEASURING AND PRICING THE RISK OF CORPORATE FAILURES*

13. Rina Hansen
*Toward a Digital Strategy for Omnichannel Retailing*

14. Eva Pallesen
*In the rhythm of welfare creation*
*A relational processual investigation moving beyond the conceptual horizon of welfare management*

15. Gouya Harirchi
*In Search of Opportunities: Three Essays on Global Linkages for Innovation*

16. Lotte Holck
*Embedded Diversity: A critical ethnographic study of the structural tensions of organizing diversity*

17. Jose Daniel Balarezo
*Learning through Scenario Planning*

18. Louise Pram Nielsen
*Knowledge dissemination based on terminological ontologies. Using eye tracking to further user interface design.*

19. Sofie Dam
*PUBLIC-PRIVATE PARTNERSHIPS FOR INNOVATION AND SUSTAINABILITY TRANSFORMATION*
*An embedded, comparative case study of municipal waste management in England and Denmark*

20. Ulrik Hartmyer Christiansen
*Follwoing the Content of Reported Risk Across the Organization*

21. Guro Refsum Sanden
*Language strategies in multinational corporations. A cross-sector study of financial service companies and manufacturing companies.*

22. Linn Gevoll
*Designing performance management for operational level*
*- A closer look on the role of design choices in framing coordination and motivation*

23. Frederik Larsen
*Objects and Social Actions – on Second-hand Valuation Practices*

24. Thorhildur Hansdottir Jetzek
*The Sustainable Value of Open Government Data Uncovering the Generative Mechanisms of Open Data through a Mixed Methods Approach*

25. Gustav Toppenberg
*Innovation-based M&A – Technological-Integration Challenges – The Case of Digital-Technology Companies*

26. Mie Plotnikof
*Challenges of Collaborative Governance An Organizational Discourse Study of Public Managers' Struggles with Collaboration across the Daycare Area*

27. Christian Garmann Johnsen
*Who Are the Post-Bureaucrats? A Philosophical Examination of the Creative Manager, the Authentic Leader and the Entrepreneur*

28. Jacob Brogaard-Kay
*Constituting Performance Management A field study of a pharmaceutical company*

29. Rasmus Ploug Jenle
*Engineering Markets for Control: Integrating Wind Power into the Danish Electricity System*

30. Morten Lindholst
*Complex Business Negotiation: Understanding Preparation and Planning*

31. Morten Grynings
*TRUST AND TRANSPARENCY FROM AN ALIGNMENT PERSPECTIVE*

32. Peter Andreas Norn
*Byregimer og styringsevne: Politisk lederskab af store byudviklingsprojekter*

33. Milan Miric
*Essays on Competition, Innovation and Firm Strategy in Digital Markets*

34. Sanne K. Hjordrup
*The Value of Talent Management Rethinking practice, problems and possibilities*

35. Johanna Sax
*Strategic Risk Management – Analyzing Antecedents and Contingencies for Value Creation*

36. Pernille Rydén
*Strategic Cognition of Social Media*

37. Mimmi Sjöklint
*The Measurable Me - The Influence of Self-tracking on the User Experience*

38. Juan Ignacio Staricco
*Towards a Fair Global Economic Regime? A critical assessment of Fair Trade through the examination of the Argentinean wine industry*

39. Marie Henriette Madsen
*Emerging and temporary connections in Quality work*

40. Yangfeng CAO
*Toward a Process Framework of Business Model Innovation in the Global Context Entrepreneurship-Enabled Dynamic Capability of Medium-Sized Multinational Enterprises*

41. Carsten Scheibye
*Enactment of the Organizational Cost Structure in Value Chain Configuration A Contribution to Strategic Cost Management*

**2016**

1. Signe Sofie Dyrby
*Enterprise Social Media at Work*

2. Dorte Boesby Dahl
*The making of the public parking attendant*
*Dirt, aesthetics and inclusion in public service work*

3. Verena Girschik
*Realizing Corporate Responsibility Positioning and Framing in Nascent Institutional Change*

4. Anders Ørding Olsen
*IN SEARCH OF SOLUTIONS*
*Inertia, Knowledge Sources and Diversity in Collaborative Problem-solving*

5. Pernille Steen Pedersen
*Udkast til et nyt copingbegreb*
*En kvalifikation af ledelsesmuligheder for at forebygge sygefravær ved psykiske problemer.*

6. Kerli Kant Hvass
*Weaving a Path from Waste to Value: Exploring fashion industry business models and the circular economy*

7. Kasper Lindskow
*Exploring Digital News Publishing Business Models – a production network approach*

8. Mikkel Mouritz Marfelt
*The chameleon workforce: Assembling and negotiating the content of a workforce*

9. Marianne Bertelsen
*Aesthetic encounters*
*Rethinking autonomy, space & time in today's world of art*

10. Louise Hauberg Wilhelmsen
*EU PERSPECTIVES ON INTERNATIONAL COMMERCIAL ARBITRATION*

11. Abid Hussain
*On the Design, Development and Use of the Social Data Analytics Tool (SODATO): Design Propositions, Patterns, and Principles for Big Social Data Analytics*

12. Mark Bruun
*Essays on Earnings Predictability*

13. Tor Bøe-Lillegraven
*BUSINESS PARADOXES, BLACK BOXES, AND BIG DATA: BEYOND ORGANIZATIONAL AMBIDEXTERITY*

14. Hadis Khonsary-Atighi
*ECONOMIC DETERMINANTS OF DOMESTIC INVESTMENT IN AN OIL-BASED ECONOMY: THE CASE OF IRAN (1965-2010)*

15. Maj Lervad Grasten
*Rule of Law or Rule by Lawyers?*
*On the Politics of Translation in Global Governance*

16. Lene Granzau Juel-Jacobsen
*SUPERMARKEDETS MODUS OPERANDI – en hverdagssociologisk undersøgelse af forholdet mellem rum og handlen og understøtte relationsopbygning?*

17. Christine Thalsgård Henriques
*In search of entrepreneurial learning – Towards a relational perspective on incubating practices?*

18. Patrick Bennett
*Essays in Education, Crime, and Job Displacement*

19. Søren Korsgaard
*Payments and Central Bank Policy*

20. Marie Kruse Skibsted
*Empirical Essays in Economics of Education and Labor*

21. Elizabeth Benedict Christensen
*The Constantly Contingent Sense of Belonging of the 1.5 Generation Undocumented Youth*
*An Everyday Perspective*

22. Lasse J. Jessen
*Essays on Discounting Behavior and Gambling Behavior*

*23. Kalle Johannes Rose*
*Når stifterviljen dør…*
*Et retsøkonomisk bidrag til 200 års juridisk konflikt om ejendomsretten*

24. Andreas Søeborg Kirkedal
*Danish Stød and Automatic Speech Recognition*

**TITLER I ATV PH.D.-SERIEN**

**1992**

1. Niels Kornum
*Servicesamkørsel – organisation, økonomi og planlægningsmetode*

**1995**

2. Verner Worm
*Nordiske virksomheder i Kina*
*Kulturspecifikke interaktionsrelationer ved nordiske virksomhedsetableringer i Kina*

**1999**

3. Mogens Bjerre
*Key Account Management of Complex Strategic Relationships*
*An Empirical Study of the Fast Moving Consumer Goods Industry*

**2000**

4. Lotte Darsø
*Innovation in the Making*
*Interaction Research with heterogeneous Groups of Knowledge Workers creating new Knowledge and new Leads*

**2001**

5. Peter Hobolt Jensen
*Managing Strategic Design Identities*
*The case of the Lego Developer Network*

**2002**

6. Peter Lohmann
*The Deleuzian Other of Organizational Change – Moving Perspectives of the Human*

7. Anne Marie Jess Hansen
*To lead from a distance: The dynamic interplay between strategy and strategizing – A case study of the strategic management process*

**2003**

8. Lotte Henriksen
*Videndeling*
*– om organisatoriske og ledelsesmæssige udfordringer ved videndeling i praksis*

9. Niels Christian Nickelsen
*Arrangements of Knowing: Coordinating Procedures Tools and Bodies in Industrial Production – a case study of the collective making of new products*

**2005**

10. Carsten Ørts Hansen
*Konstruktion af ledelsesteknologier og effektivitet*

**TITLER I DBA PH.D.-SERIEN**

**2007**

1. Peter Kastrup-Misir
*Endeavoring to Understand Market Orientation – and the concomitant co-mutation of the researched, the re searcher, the research itself and the truth*

**2009**

1. Torkild Leo Thellefsen
*Fundamental Signs and Significance effects*
*A Semeiotic outline of Fundamental Signs, Significance-effects, Knowledge Profiling and their use in Knowledge Organization and Branding*

2. Daniel Ronzani
*When Bits Learn to Walk Don't Make Them Trip. Technological Innovation and the Role of Regulation by Law in Information Systems Research: the Case of Radio Frequency Identification (RFID)*

**2010**

1. Alexander Carnera
*Magten over livet og livet som magt*
*Studier i den biopolitiske ambivalens*