Monotone String-to-String Translation for NLU and ASR Tasks

Von der Fakultät für
Mathematik, Informatik und Naturwissenschaften der
RWTH AACHEN UNIVERSITY

zur Erlangung des akademischen Grades eines
Doktors der Naturwissenschaften
genehmigte Dissertation

vorgelegt von
Diplom-Informatiker Stefan Hahn
aus Prüm

Berichter:
Univ.-Prof. Dr.-Ing. Hermann Ney
Prof. em. Dr. Renato De Mori

Tag der mündlichen Prüfung: 13. November 2014

Dipl.-Inform. Stefan Hahn
Human Language Technology and Pattern Recognition Group
RWTH Aachen University
hahn@cs.rwth-aachen.de

Erklärung

Hiermit versichere ich, dass ich die vorliegende Doktorarbeit selbstständig verfasst und keine anderen als die angegebenen Hilfsmittel verwendet habe. Alle Textauszüge und Grafiken, die sinngemäß oder wörtlich aus veröffentlichten Schriften entnommen wurden, sind durch Referenzen gekennzeichnet.

Aachen, den 17.02.2014

Dipl.-Inform. Stefan Hahn

Abstract

Monotone string-to-string translation problems have to be tackled as part of almost all state-of-the-art natural language understanding and large vocabulary continuous speech recognition systems. In this work, two such tasks will be investigated in detail and improved using conditional random fields, namely concept tagging and grapheme-to-phoneme conversion.

Concept tagging is usually one of the first modules within a dialogue or natural language understanding system. Here, the recognition result of a speech recognition system is augmented with task and domain dependent semantic information. Within this work, six different approaches are compared and evaluated on three different tasks in various languages on several levels. Considered are manual transcriptions versus speech recognition hypotheses as input as well as attribute name and attribute value level tags as output. By using an improved approach based on conditional random fields, the best results on all tasks and languages could be achieved. On the well-known French MEDIA task, conditional random fields lead to a concept error rate of 12.6% for attribute name and value extraction, which is a 35% relative improvement over the best published result within the MEDIA evaluation campaign in 2005 in the relaxed-simplified condition with 19.6%.

The improvements over the classical conditional random fields-based approach as for example the introduction of a modified training criterion are discussed in detail. Additionally, recognizer output voting error reduction is applied as a system combination technique which could further reduce the concept error rate. A combination of rule-based and statistical attribute value extraction based on conditional random fields could be developed to improve over the standard rule-based baseline.

The second monotone string-to-string translation task covers grapheme-to-phoneme conversion. Here, the pronunciation of a given word is derived automatically. With such a conversion module, it is possible to augment pronunciation dictionaries for speech recognition with e.g. named entities or other domain specific words, which might change over time. From a conceptual point, the main difference between this task and concept tagging is that an alignment between source and target side has to be modelled or given.

In a first series of experiments, various state-of-the-art generative grapheme-to-phoneme conversion approaches are compared and evaluated on large pronunciation dictionaries in various languages. For the application of conditional random fields, a number of features and techniques to reduce computational complexity had to be implemented and derived. The alignment problem has been tackled by either using an external model or integrating a hidden variable within the conditional random fields training process. Using these modifications, state-of-the-art accuracy results could be achieved on a couple of English pronunciation dictionaries.

Additionally, state-of-the-art speech recognition systems have been trained using a grapheme-to-phoneme conversion module based on hidden conditional random fields and compared with

speech recognition systems where a joint-n-gram approach has been used to provide pronunciations for words which are not part of the background lexicon. In an extensive comparison across several test sets from the English QUAERO tasks, the word error rate for speech recognition systems utilizing hidden conditional random fields could outperform the systems using the generative joint-n-gram based approach by 1–3% relatively. Note that the automatic speech recognition systems only differ by the grapheme-to-phoneme conversion system.

In summary, for both tasks considered in this thesis, methods based on (hidden) conditional random fields could be derived outperforming state-of-the-art approaches.

Zusammenfassung

Innerhalb annähernd aller heutigen Aufgaben in den Bereichen des automatischen Sprachverstehens als auch der automatischen Spracherkennung spielen monotone Wort-zu-Wort Übersetzungsprobleme eine große Rolle. In dieser Arbeit werden zwei dieser Probleme (Konzept-Tagging und Graphem-zu-Phonem Konvertierung) näher untersucht und die Übersetzungsqualität mittels sogenannter Conditional Random Fields verbessert.

Konzept-Tagging ist üblicherweise eines der ersten Module innerhalb eines Dialog-Systems oder eines Systems zum Sprachverstehen. Zur Informationsextraktion wird hier das Erkennungsergebnis eines Spracherkennungssystems mit aufgaben- und domänenabhängiger semantischer Information angereichert. Sechs verschiedene Ansätze zur Lösung des Konzept-Tagging Problems werden in dieser Arbeit miteinander verglichen und auf drei verschiedenen Aufgaben und Sprachen auf verschiedenen Ebenen experimentell bewertet. Betrachtet werden sowohl der Unterschied in der Performanz zwischen manueller Transkription im Vergleich zu Hypothesen generiert von einem automatischen Spracherkenner als Eingabe als auch die Auswertung auf der Ebene von Attributnamen und Attributwerten. Unabhängig von der Aufgabe und der Sprache führen Ansätze basierend auf Conditional Random Fields zu den besten Ergebnissen. Auf der bekannten französischen MEDIA Aufgabe konnte mit Hilfe dieser Technik eine Konzeptfehlerrate von 12.6% erreicht werden. Dies entspricht einer Verbesserung von 35% relativ gegenüber der besten, publizierten Fehlerrate innerhalb der MEDIA Evaluierungskampagne von 2005 in der sogenannten "relaxed-simplified" Bedingung (19.6%).

Zusätzlich zu Verbesserungen gegenüber dem klassischen Conditional Random Fields Ansatz, z.B. ein modifiziertes Trainingskriterium, werden Systemkombinationsergebnisse mittels der sogenannten ROVER Methode (recognizer output voting error reduction) vorgestellt, welche nochmals die Konzeptfehlerrate reduzieren konnten. Ferner wurde eine Kombination von regelbasierter und statistischer Attributwerteextraktion entwickelt, durch die Verbesserungen gegenüber den regelbasierten Ausgangswerten erzielt werden konnten.

Das zweite Problem aus dem Bereich der Wort-zu-Wort Übersetzungen beschäftigt sich mit der Graphem-zu-Phonem Konvertierung. Ziel ist es, die Aussprache eines gegebenen Wortes automatisch zu bestimmen. Mittels eines solchen Konvertierungsmoduls kann ein Aussprachelexikon eines automatischen Spracherkenners mit z.B. Eigennamen oder domänenspezifischen Wörtern ergänzt werden, welche sich auch im Laufe der Zeit ändern können. Von einem konzeptuellen Standpunkt aus gesehen, ist der Unterschied zwischen dieser und der Konzept-Tagging Aufgabe der, dass eine Alignierung zwischen Graphemen und Phonemen entweder vorgegeben oder zusätzlich modelliert werden muss.

Im ersten experimentellen Teil zur Grapheme-zu-Phonem Konvertierung werden verschiedene generative Ansätze zur Lösung dieses Problems verglichen und experimentell auf großen Aussprachelexika in verschiedenen Sprachen ausgewertet. Um Conditional Random Fields

erfolgreich anzuwenden, bedurfte es der Implementierung und Herleitung einer Reihe von Modifikationen und Techniken, um die Rechenintensität der Algorithmen zu reduzieren. Das Alignierungsproblem wurde dadurch bewältigt, dass entweder ein externes Modell zur Bestimmung des Alignments eingesetzt wurde oder das Problem direkt innerhalb der Conditional Random Fields mit Hilfe einer verstecken Variable integriert wurde (Hidden Conditional Random Field). Mit Hilfe dieser Modifikationen konnten auf einigen englischen Aussprachelexika Ergebnisse erzielt werden, die dem heutigen Stand der Technik entsprechen.

In einem zweiten experimentellen Teil wurde ein Modul zur Graphem-zu-Phonem Konvertierung mittels Hidden Conditional Random Fields trainiert und innerhalb eines automatischen Spracherkenners verwendet. Die Ergebnisse wurden mit einem Graphem-zu-Phonem Konvertierungsmodul verglichen, welches mittels sogenannten zusammengefügten n-grammen trainiert wurde (joint n-grams), was dem de-facto Standard entspricht. Mit beiden Methoden wurden Wörter, die nicht im Hintergrundlexikon waren, phonetisiert und dem Erkennerlexikon hinzugefügt. In einem umfassenden Vergleich auf verschiedenen Testkorpora aus den englischen QUAERO Aufgaben ist die Wortfehlerrate von Spracherkennungssystemen mit einem auf Hidden Conditional Random Fields-basierten Graphem-zu-Phoneme Konvertierungsmodul um 1–3% kleiner als mit einem generativen joint-n-gram Ansatz. Die Spracherkennungssysteme unterscheiden sich dabei lediglich um das Graphem-zu-Phonem Konvertierungsmodul.

Zusammenfassend konnten für beide betrachteten Probleme Methoden basierend auf (Hidden) Conditional Random Fields entwickelt und angewendet werden, die den aktuellen Stand der Technik übertreffen.

Acknowledgment

About seven years ago, I started to work as a research assistant at the Human Language Technology and Pattern Recognition Group at RWTH Aachen University. It was a fantastic ride and I would like to thank all the people who supported me and this work throughout the last years.

First, I would like to thank Prof. Dr.-Ing. Hermann Ney for giving me the opportunity to work on my PhD thesis at his department. I am grateful for the freedom he gave me in choosing my own research directions. I am also very grateful that Prof. em. Dr. Renato de Mori agreed to review my thesis. We already worked together within the LUNA project, where he was the technical coordinator and inspired me by his calm manner and clear vision. Thank you, Renato.

Furthermore, I would like to thank Dr. rer. nat. Ralf Schlüter for his support related to ASR questions, especially in my early days.

A special thanks goes to Patrick Lehnen. We worked together in the LUNA project resulting in large parts of this thesis. He designed and wrote most parts of the CRF framework which has been used for many of the presented experiments. Together with Jesús Andrés-Ferrer, he also proofread this work.

The past years would not have been as much fun without my office mate and friend David Rybach. I could always rely on his expertise in programming as well as relaxing coffee breaks.

I would like to thank all of my colleagues and friends in the speech recognition group for many helpful discussions as well as collaborations in evaluations and writing papers. This includes in no particular order Pavel Golik, Christian Gollan, Georg Heigold, Björn Hoffmeister, Jonas Lööf, Amr El-Desoky Mousa, David Nolden, Markus Nußbaum-Thom, Christian Oberdörfer, Christian Plahl, Muhammad Ali Tahir, Zoltan Tüske, Basha Shaik, Martin Sundermeyer, Simon Wiesler.

Since parts of this thesis are concerned with machine translation, I had the pleasure to also work with and learn from my colleagues and friends from the machine translation group including Oliver Bender, Jan Bungeroth, Markus Freitag, Andy Guta, Saša Hasan, Carmen Heger, Matthias Huck, Sharam Khadivi, Gregor Leusch, Saab Mansour, Evgeny Matusov, Arne Mauser, Malte Nuhn, Stephan Peitz, Jan-Thorsten Peter, Maja Popović, Christoph Schmidt, Daniel Stein, David Vilar, Jörn Wübker, Jia Xu, Yuqi Zhang.

Some collaboration efforts touched the realms of image processing, particularly handwriting recognition. I would like to thank the respective colleagues for their support, including Thomas Deselaers, Philippe Dreuw (thanks for the great LaTeX templates), Jens Forster, Michal Kozielski.

I would also like to thank Paul Vozila and Max Bisani for the great time I had during my internship at Nuance in Burlington and the many things I learned.

Finally, I would like to thank my parents from the bottom of my heart for their support throughout my studies.

This work was partly funded by the European Union under the specific targeted research project LUNA - spoken language understanding in multilingual communication systems (FP6-033549) and this work was partly realized as part of the Quaero Programme, funded by OSEO, French State agency for innovation.

Outline

Chapter 1

Introduction

In this thesis, various approaches to tackle monotone string-to-string translation tasks are presented and compared. String-to-string translation should be understood as the translation of a source token sequence into a target token sequence, whereas these tokens do not necessarily have to be natural words. Examples for such tasks include e.g. the translation of higher-level programming languages into machine code by a compiler or the translation of a source language to a target language, as would be the case for statistical machine translation (SMT). Additionally, we restrict ourselves to problems where a monotone translation suffices. Here, tokens are not re-ordered during search, i.e. if a token is skipped at a certain position, it will not be inserted later on. In other words, there exists a monotone mapping from source tokens to target tokens, defined by an alignment.

Two tasks from two domains form the motivation and the focus for our work. On the one hand, the problem of assigning meaning to a natural sentence in form of concepts (*concept tagging*) is considered. Within a state-of-the-art dialogue system, this is a crucial component. Due to the recent occurrence of more and more applications for natural language understanding (NLU), to improve the performance of such methods is important. On the other hand, a similar monotone string-to-string translation problem has to be solved to build a state-of-the-art automatic speech recognition (ASR) system. Especially when no linguistic expert is available or deemed to be to expensive, automatic systems are needed to derive pronunciations for words which are so far unknown to the speech recognizer. The *grapheme-to-phoneme conversion (G2P)* task is thus also important to improve the performance of ASR systems, especially in applications where an end user may add words to a recognition system. End user customization is more and more common, especially on personalized devices as home computers or smartphones.

Both tasks are already well understood and research is being conducted by many groups for many years. A detailed presentation of related work is presented in Chapter 3. Since both, concept tagging as well as G2P are parts of larger systems, introductions to these systems are given in this section. This includes an introduction to SMT to better understand the similarities and differences to the monotone translation tasks which are the focus of this work. The introduction to SMT is given in Section 1.1. In Section 1.2 follows an introduction to statistical NLU, which illustrates the framework in which a concept tagging module is usually applied. The concept tagging task will be presented in detail.

A standard approach to automatic speech recognition will be presented in Section 1.3. This includes a detailed presentation of the G2P task. But since the input to a dialogue manager is

usually given by an ASR system, this section is also relevant to better understand the concept tagging task and the motivation to derive tagging methods to cope with erroneous speech input.

In the literature, mostly generative models have been used to tackle the aforementioned monotone string-to-string translation tasks. Besides an in-depth comparison of well-known generative methods, we will present discriminative or log-linear approaches to tackle these tasks, most notably conditional random fields (CRFs). The respective theoretical background is given in Section 1.4.

We evaluated and compared the various methods to solve the concept tagging task on various state-of-the-art NLU and spoken language understanding (SLU) corpora in various languages and could show that CRFs outperform all other tested approaches. With respect to the G2P task, we could show that CRFs lead to state-of-the-art results on various English pronunciation dictionaries and that it is possible to improve an ASR system by using CRFs for the G2P component.

Section 1.5 gives an overview of the structure of this document and concludes the introduction.

1.1 Machine Translation

The translation of a given (human) sentence in a source language into a (human) sentence in another target language conveying the same meaning as the original sentence is the goal of machine translation (MT). First attempts to tackle this task in an automatic manner have been considered already in the late forties (cf. a later reprint in [Weaver 55]) and early fifties [Bar-Hillel 51, IBM 54], especially after the highly influential work on the source channel approach on communication has been published, founding the field of information theory [Shannon 48]. There are many possibilities to model the MT task and it is not easy to classify all of those. A well-known, high-level classification has been proposed in [Vauquois 68] and is depicted in Figure 1.1.

Here, three variants can be distinguished. Within dictionary-based MT, a direct word-to-word translation via dictionary look-up is performed. The transfer-based translation approach does additionally consider a possibly different structure of the two languages. Morphology and syntax of the source sentences is analyzed and the sentences are transformed into an internal representation which is then translated via bilingual dictionary look-ups. Within the theoretical interlingua MT approach, the source language would be transformed even further, namely into a language-independent representation, referred to as *interlingua*, which would then be translated into the target language. Such an interlingua has not been found so far.

There are now in principle two ways of how to generate the target text out of a given source text, namely *rule-based* and *data-driven*. Within rule-based approaches, a set of rules describing the translation process is formulated by linguistic experts. While this approach leads to good results, it is very costly to derive and maintain the rules as well as time consuming. Additionally, it might be difficult to adapt existing rules for new tasks / domains, since it might be hard to keep track of the rule interactions. More details and discussions about rule-based MT can be found e.g. in [Hutchins & Somers 92, Arnold & Balkan$^+$ 94, Lagarda & Alabau$^+$ 09]. In

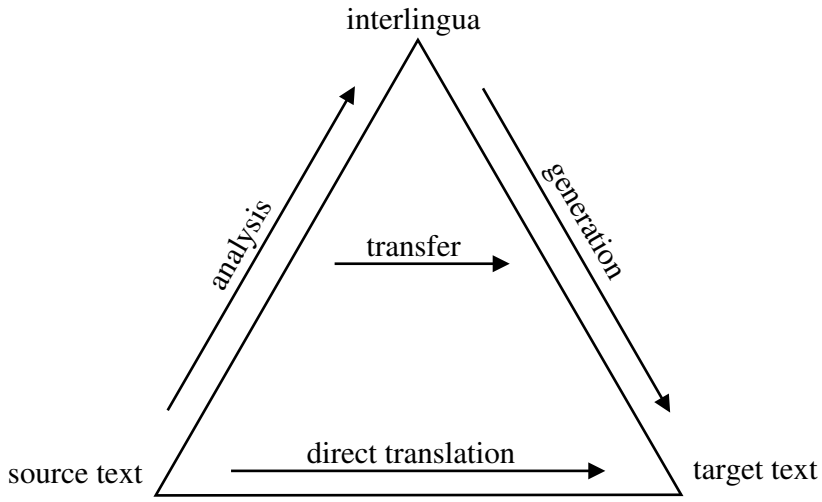


Figure 1.1 Pyramid diagram of translation methods after [Vauquois 68]. The varying degree of linguistic analysis for machine translation is depicted from the direct word-to-word translation, via a transfer-based translation involving e.g. morphology and syntax analysis, to a translation using a language independent *interlingua*.

our work, we are focussing on data-driven approaches, more precisely on *statistical machine translation*. Here, large corpora of mono- and bilingual text are used as knowledge sources to build a (or several) statistical model(s). Given a sentence in a source language, this model(s) are used to derive the most likely translation. The mathematical background is presented in the next subsection. Although not used in this work, it should be noted that there is also an approach known as *example-based translation* or translation by analogy. Here, from the bilingual corpora, examples are generated which are pieced together to form the final translated sentence in the target language.

1.1.1 Statistical Machine Translation

Although research already started in the fifties, statistical machine translation (SMT) only took off in the late eighties / early nineties when International Business Machines Corporation (IBM) introduced the later so-called IBM-models [Brown & Cocke$^+$ 88, Brown & Cocke$^+$ 90, Brown & Della Pietra$^+$ 93]. The (original) problem formulation is as follows: given a French source sentence $f_1, \ldots, f_J$, which English target sentence $e_1, \ldots, e_I$ do we chose? Using the maximum a posteriori (MAP) decision rule, we get

$$\hat{e}_i^{\hat{I}} = \underset{I, e_i^I}{\operatorname{argmax}} \left\{ Pr(e_1^I | f_1^J) \right\} \tag{1.1}$$

Here, $\hat{e}_i^I$ denotes the best hypothesis, i.e. the one with the highest probability. Within the aforementioned work by IBM, this probability is further broken down using Bayes' decision rule:

$$\hat{e}_i^{\hat{I}} = \underset{I,e_i^I}{\operatorname{argmax}} \left\{ \frac{Pr(e_1^I) \cdot Pr(f_1^J|e_1^I)}{Pr(f_1^J)} \right\} \tag{1.2}$$

$$= \underset{I,e_i^I}{\operatorname{argmax}} \left\{ Pr(e_1^I) \cdot Pr(f_1^J|e_1^I) \right\} \tag{1.3}$$

The resulting Equation 1.3 is referred to as the source-channel approach to SMT [Brown & Cocke$^+$ 90] and the basis for many modeling approaches, decomposing the original posterior probability $Pr(e_1^I|f_1^J)$ into a language model on target side $Pr(e_1^I)$ and a translation model $Pr(f_1^J|e_1^I)$. The IBM models 1–5 now define a structured way to train the translation model. IBM model 1 is trained from scratch and is used to initialized IBM model 2 and so on. Additionally to the IBM models, the so-called hidden Markov model (HMM) is often used [Vogel & Ney$^+$ 96]. Within this work, especially the IBM model 1 will be used to initialize CRF models with hidden alignments (cf. Section 8.6).

More recently, a log-linear decomposition of the translation model has been proposed, which is usually used in today's SMT systems [Och & Ney 02]:

$$Pr(e_1^I|f_1^J) = \frac{\exp\left(\sum_{m=1}^M \lambda_m h_m(f_1^J, e_1^I)\right)}{\sum_{\tilde{e}_1^I} \exp\left(\sum_{m=1}^M \lambda_m h_m(f_1^J, \tilde{e}_1^I)\right)} \tag{1.4}$$

Here, the h_m define feature functions, which usually are various statistical models. These models are weighted with λ_m, which are usually optimized using minimum error rate training (MERT) [Och 03]. If we use the logarithm of the language model on target side and the translation model from Equation 1.3 as feature functions h_1 and h_2 weighted with $\lambda_1 = \lambda_2 = 1$, we get the source-channel approach as a special case of the more flexible and more general log-linear framework. Concerning search, we can now derive the following equation for the best hypothesis $\hat{e}_1^{\hat{I}}$:

$$\hat{e}_1^{\hat{I}} = \underset{I,e_1^I}{\operatorname{argmax}} \left\{ \frac{\exp\left(\sum_{m=1}^M \lambda_m h_m(f_1^J, e_1^I)\right)}{\sum_{\tilde{e}_1^I} \exp\left(\sum_{m=1}^M \lambda_m h_m(f_1^J, \tilde{e}_1^I)\right)} \right\} \tag{1.5}$$

$$= \underset{I,e_1^I}{\operatorname{argmax}} \left\{ \sum_{m=1}^M \lambda_m h_m(f_1^J, e_1^I) \right\} \tag{1.6}$$

Starting from Equation 1.5, Equation 1.6 is derived by dropping the normalization term, since it does not affect the maximization process, as well as the monotone exponential function. It should be noted that the normalization term given in Equation 1.4 is rarely used in practise. Figure 1.2 shows a typical SMT system design built upon the log-linear framework. Within the figure and in the presented SMT approach, the feature functions are represented as a number of statistical models including phrase-based models in source-to-target and target-to-source direction, IBM-1 like scores at phrase level, again in source-to-target and target-to-source direction, a target language model, and additional word and phrase penalties. We do also incorporate this

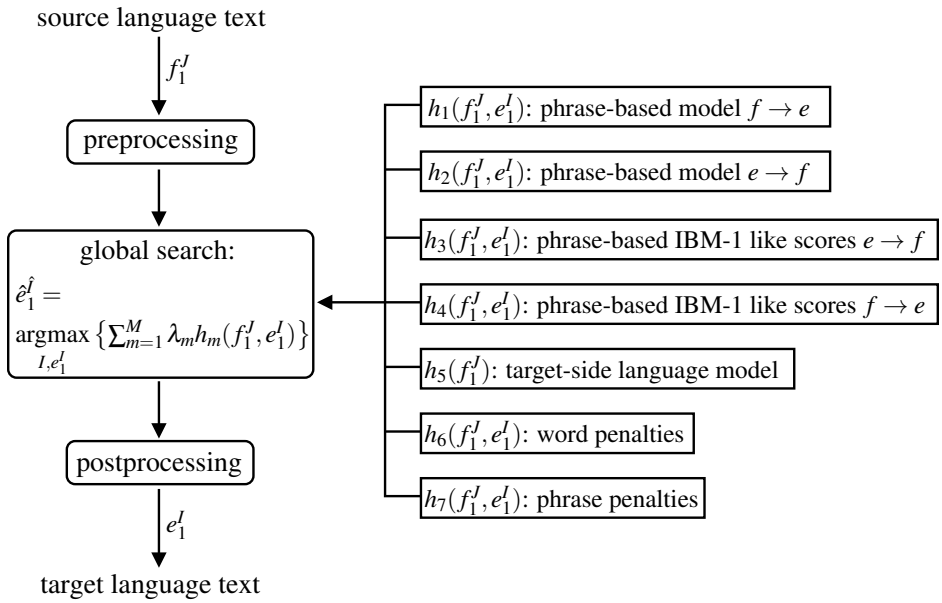


Figure 1.2 Diagram of a typical machine translation system following the log-linear modeling approach. Here, seven generative models are used and log-linearly combined for the global search process.

SMT model within a comparison of concept tagging approaches in Section 4.1.4. IBM 1 scores are used to initialize hidden conditional random field (HCRF) models as depicted in Section 8.6. A more detailed description of the whole SMT system can be found in [Mauser & Zens+ 06].

The phrase-based models have been introduced in [Och & Tillmann+ 99, Zens & Och+ 02, Koehn & Och+ 03] to overcome the limitations of the IBM models, which only rely on a word-to-word alignment, i.e. disregarding context information. Phrases are here defined as word sequences without any further linguistic or semantic meaning, and they are automatically derived. The phrase-generation usually starts with a given word-alignment obtained from the IBM models and for any extracted bilingual phrase pair $\langle \tilde{f}, \tilde{e} \rangle$ holds that $\tilde{f}$ and $\tilde{e}$ are contiguous, non-empty word sequences in the source and target language respectively and words in $\tilde{f}$ are only aligned to words in $\tilde{e}$ and vice versa. In Figure 1.3, an example alignment between an English source and a French target sentence is presented with the extracted phrase-pairs marked with boxes.

The French words “de” and “la” are not aligned to any English words, since they have no counterparts in the English sentence. Additionally, only the largest phrases are shown without any sub-phrases.

The alignment/segmentation of the source and target sentence is usually integrated within the search process as a hidden variable A, starting from a log-linear approach as presented in Equation 1.4:

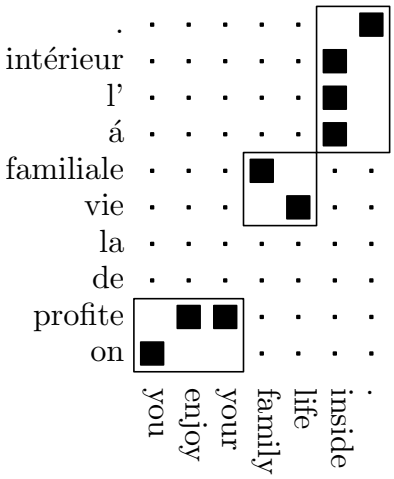


Figure 1.3 An example alignment from a phrase-based translation system. The phrases are marked with boxes, whereas aligned words are denoted with a black mutton. Note that not all possible phrases are shown, but only the largest ones. The French words "de" and "la" are not aligned to any English words.

$$Pr(e_1^I|f_1^J) = \sum_A Pr(e_1^I, A|f_1^J) \tag{1.7}$$

$$= \sum_A \frac{\exp\left(\sum_{m=1}^M \lambda_m h_m(e_1^I, A; f_1^J)\right)}{\sum_{e'^{I'}_1, A'} \exp\left(\sum_{m=1}^M \lambda_m h_m(e'^{I'}_1, A'; f_1^J)\right)} \tag{1.8}$$

$$\approx \max_A \frac{\exp\left(\sum_{m=1}^M \lambda_m h_m(e_1^I, A; f_1^J)\right)}{\sum_{e'^{I'}_1, A'} \exp\left(\sum_{m=1}^M \lambda_m h_m(e'^{I'}_1, A'; f_1^J)\right)}. \tag{1.9}$$

Here, the maximum approximation is usually applied (cf. Equation 1.9). The decision rule from Equation 1.6 is consequently modified accordingly:

$$\hat{e}_1^{\hat{I}} = \underset{I, e_1^I}{\operatorname{argmax}} \left\{ \max_A \sum_{m=1}^M \lambda_m h_m(e_1^I, A; f_1^J) \right\} \tag{1.10}$$

There have been many improvements to the phrase-based translation approach, most notably the hierarchical phrased-based translation [Chiang 05, Chiang 07], but the discussion would go beyond the scope of this work. For an extensive overview about SMT related literature, the reader is referred to the website maintained by Phillipp Koehn [Koehn 13] as well as the Machine Translation Archive maintained by John Hutchins [Hutchins 13]. Additionally, an overview about SMT in general is given in [Koehn 10].

1.1.2 Monotone Translation

Within the previous section, one important part of any state-of-the-art SMT system has been spared, namely the re-ordering. As can be seen in Figure 1.3, depending on the language pair, certain parts of the sentences have to be re-ordered as a preprocessing step to allow for a good alignment and thus a good translation. In the example, the English phrase "family life" corresponds to the French "vie familiale", which does brake monotonicity.

Re-ordering has already been used by the first word-based IBM approaches [Berger & Brown$^+$ 96] and is an ongoing field of SMT research [Tillmann & Ney 03, Zens & Ney 03, Tomás & Casacuberta 04, Kumar & Byrne 05, Birch & Blunsom$^+$ 09]. It has a high impact on the complexity for an SMT system, since it is NP-hard [Knight 99].

Within this work, however, we will only deal with *monotone* translation problems, more specifically *concept tagging* and *grapheme-to-phoneme conversion*, which by definition do not need any re-ordering of either source or target sentences w.r.t. the alignment. Additionally, at least for the concept tagging task which will be discussed in the context of natural language understanding (NLU) in the next section, a one-to-one alignment does naturally exist between the input words and the concept tags, since concept tags can be seen as an artificial language which is designed in such a way. This alignment is usually shipped with the corpus. The two considered monotone translation tasks are introduced in the following sections.

1.2 Natural Language Understanding

Roughly speaking, the interpretation of spoken (human) language is the focus of natural language understanding (NLU). So-called spoken language understanding (SLU) systems are used to accomplish this task. These systems are built from multiple components. A possible decomposition is shown in Figure 1.4. First, the human speech has to be transcribed automatically using an ASR system, which provides the single-best recognition sequence or a word lattice to the SLU system. Since the ASR system by itself is a complex system which needs to be understood at least roughly, Section 1.3 provides a short introduction. The understanding system is again composed of various modules which aim at enriching the raw word hypotheses with semantic structure described by a meaning representation language (MRL). The MRL is typically based on linguistic theories as e.g. presented in [Jackendoff 90]. Fragments of the SLU system application's ontology which form semantic constituents are composed to form theses semantic structures.

The first component (*concept tagging*) within the example SLU system enriches the ASR hypotheses with meaning units referred to as *concepts*. Since this module is the focus of the work presented in this thesis w.r.t. NLU, more details will be given in the following section. Relations between these smallest units of meaning are inferred and annotated by the second module, *semantic composition* (cf. e.g. [Duvert & Meurs$^+$ 08, Meurs & Lefèvre$^+$ 09]). The third and last module performs *context sensitive validation*. Here, contextual information from other modules of the dialog system (mainly the dialog manager) are taken into account to rescore semantic composition hypotheses. Other tasks of this module might be to detect wrongly directed calls within call-center applications or the detection of prank calls. Additionally, this

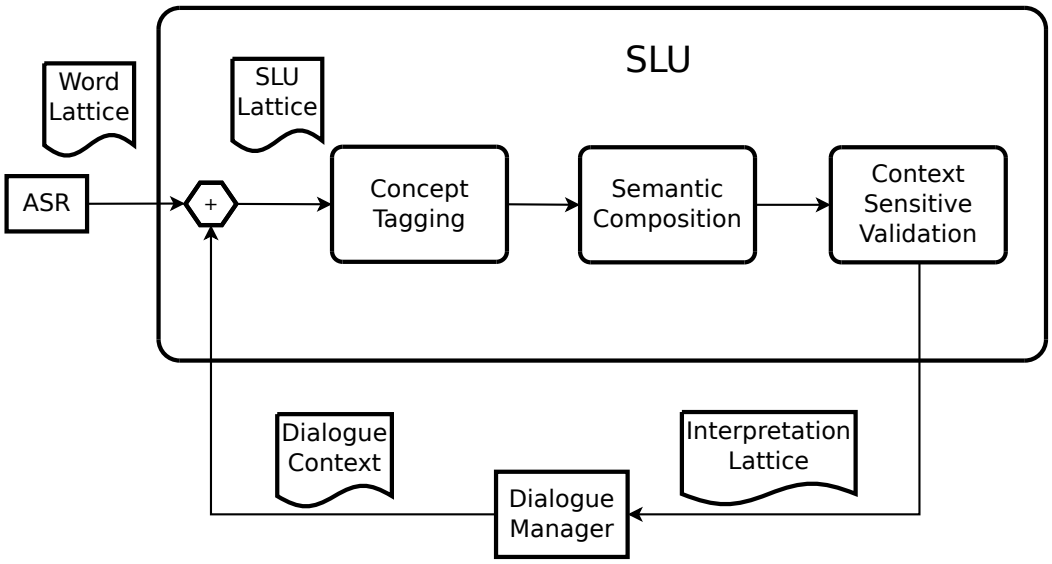


Figure 1.4 Composition and data flow a typical spoken language understanding system as used within the European LUNA project.

module might be used for coreference resolution (cf. e.g. [van Deemter & Kibble 00]) or to adapt to user behavior using online learning strategies as e.g. presented in [Damnati & Béchet+ 07]. The resulting interpretation lattice is then forwarded to the *dialog manager*. Depending on the dialog context, various actions may be required like e.g. to ask the user for additional information or to proceed within the dialog (depending on the task/application). Usually, the dialog manager does not perform any language processing and is thus not considered a part of the SLU system itself.

Since automatic transcriptions of speech usually contain errors introduced by the ASR system, interpretation of these transcriptions is particularly difficult. Especially rule-based or grammar-based systems are likely to fail on erroneous input. To take the effect of errors on the quality of transcription into account, probabilistic interpretation methods have been introduced.

Since we are mostly interested in the concept tagging module, the reader is referred to the literature for more details on the various other parts of an SLU system.

1.2.1 Concept Tagging

Concept tagging is defined as the process of extracting smallest units of meaning out of a given input word sequence. More formally, the task can be described as the annotation of a sequence of words $w_1^N = w_1 \dots w_N$ with a sequence of concepts $c_1^N = c_1 \dots c_N$. For the sake of consistency, we will stick to the following naming scheme: a *concept* should be understood as a set of attributes which is assigned to a sequence of words. This set contains up to two elements: the attribute name and the attribute value. Here, the *attribute name* tag represents the semantic meaning of the word sequence and is required for each concept. Depending on the attribute

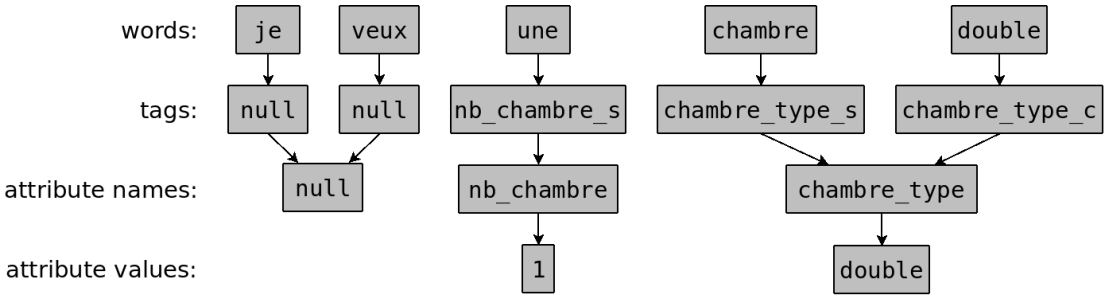


Figure 1.5 Example illustrating the general idea of concept tagging (French: "I would like a double-bed room"). The first line shows the input word sequence, the third and fourth line the appropriate attribute names and values. The second line shows how the one-to-one alignment is modelled using "start" (s) and "continue" (c) tags.

name, the *attribute value* represents an associated normalized value which has to be extracted additionally from the word sequence. An attribute value is not necessarily part of a concept.

An example form the French MEDIA corpus illustrating the task of concept tagging and the distinction between attribute name and value is given in Figure 1.5.

Four layers are depicted. The input word sequence is shown in the first line, the resulting attribute names and accompanying values are shown in lines three and four. Here, the attribute value accompanying the **nb_chambre** ("number of rooms") attribute name is reflected by an integer number, whereas the attribute value related to the type of room (**chambre_type**) is denoted by a string. Line two depicts the so-called begin insides outside scheme (BIO) notation (cf. [Ramshaw & Marcus 95]) on attribute name tag level, which is used to obtain a one-to-one alignment between words and attribute names. For example, the utterance part `chambre double` is mapped from the attribute name

$$\underbrace{\textbf{chambre-type}\{\texttt{chambre double}\}}_{c_3{:}w_4,w_5}$$

to the two attribute name / concept *tags*

$$\underbrace{\textbf{chambre-type\_s}\{\texttt{chambre}\}}_{t_4{:}w_4}\underbrace{\textbf{chambre-type\_c}\{\texttt{double}\}}_{t_5{:}w_5}$$

using the start/continue scheme. A disadvantage of applying this mapping scheme is that the number of output symbols is roughly doubled.

Some more details about the concept tagging task itself can be found in Chapter 4, e.g. more details on the one-to-one alignment are presented in Section 4.1.1 whereas another example from the earlier US-English Air Travel Information System (ATIS) task is presented in Figure 4.1.

As already denoted in the previous section, SLU and thus concept tagging is difficult because errors are introduced by the ASR process. To tackle this task, several statistical approaches have been proposed in the literature, e.g. conceptual HMMs are proposed in the Chronus system [Pieraccini & Levin$^+$ 91]. Within this approach, concepts are introduced as hidden states whereas the observations are the ASR word hypotheses.

Another proposition to tackle the task of concept tagging is based on stochastic grammars as presented in [Seneff 89, Miller & Schwartz+ 94]. Since spoken language does not always follow the rules of a manually defined, formal grammar, it is difficult to obtain correct parse trees from the erroneous ASR hypotheses. Thus, partial parsing has been considered. Here, fragments of the application ontology have been combined to form semantic constituents. They can be annotated with word sequences of finite length. Finite state transducers (FSTs) have then been built using these constituent annotations. Using this model, it is possible to annotate a word sequence with e.g. part-of-speech tags like noun phrases. Hereby, the length of the tagged word sequence might be variable and possibly long.

In general, with respect to the concept tagging task, two types of statistical approaches can be distinguished which are often used. On the one hand, *generative* approaches model the joint probability $P(w_1^N, c_1^T)$ of a sequence of words $w_1, \ldots, w_N$ and a sequence of concepts $c_1, \ldots, c_T$. Thus, they are able to generate samples from the joint distribution. Within the work presented in this paper, dynamic Bayesian networks (DBNs), and FSTs are within this category. Additionally, the SMT approach can also be considered to be a generative one, although the final hypotheses are derived by a log-linear (discriminative) combination of several generative models. Another possibility is to consider *discriminative* classification approaches, which model the conditional probability distribution $P(c_1^T|w_1^N)$ directly. Within this work, support vector machines (SVMs), maximum entropy Markov models (MEMMs) and CRFs belong to this class of models. More details on the various models can be found in the respective sections in Chapter 4. Since one focus of this work are CRFs, they will be introduced together with the closely related MEMMs in detail in Section 1.4. In the attempts to combine features from generative and discriminative models, exponential models have also been considered and evaluated. Some of them are used in SMT to go from natural language to a constituent MRL improving early approaches proposed in [Papineni & Roukos+ 98]. In general, also discriminative models like MEMMs and CRFs might be used to combine generative and discriminative models. Note that it has been shown that generative models can be converted into discriminative models, and more importantly also vice versa, at least in principle [Heigold & Lehnen+ 08].

Some of the generative and discriminative approaches compared and evaluated in this work have been compared in the literature before, e.g. in [Rubinstein & Hastie 97, Santafé & Lozano+ 07, Raymond & Riccardi 07, De Mori & Hakkani-Tur+ 08]. In the first reference it is concluded that discriminative training is expensive albeit more robust and special knowledge about the true distribution is not needed. In contrast, training of generative approaches is cheap but the models need to fit well to the true distribution.

More pointers to literature with respect to SLU and particularly concept tagging are given in Section 3.1.

1.3 Automatic Speech Recognition

Speech has been used by humans to communicate much earlier than written language and is thus one of the most natural ways of communication. To allow for a speech-based human-computer interface is the main goal of automatic speech recognition (ASR). While first appli-

cations where just targeted at recognizing a very limited number of words (e.g. the ten digits $0,\ldots,9$) in controlled acoustic conditions and tailored to one speaker (cf. [Davis & Biddulph+ 51]), ASR is nowadays deployed successfully commercially in many areas, e.g. dictation systems for personal computers, dictation solutions for the health care and legal sector, products to support disabled persons, voice controlled in-car applications or search applications for mobile devices, often combined with NLU or a translation module, just to name a few.

More formally, the task of automatic speech recognition can be described as a translation from the acoustic signal into valid words within the respective language. Most state-of-the-art ASR systems are based on statistical models. The resulting large vocabulary continuous speech recognition (LVCSR) systems are able to distinguish between tenths of thousands of words and are usually build by combining several models representing various knowledge sources. The details are now presented in the following section.

1.3.1 Statistical Speech Recognition

Within the statistical approach to speech recognition, the problem of finding the best word sequence $\hat{w}_1^N$ given a sequence of acoustic features x_1^T is usually solved by maximizing the posterior probability broken down using Bayes' decision rule [Bayes 63] and very similar to the approach adopted later by SMT and presented in Equation 1.1:

$$\hat{w}_1^N = \underset{N,w_1^N}{\operatorname{argmax}} \left\{ Pr(w_1^N|x_1^T) \right\} \tag{1.11}$$

$$= \underset{N,w_1^N}{\operatorname{argmax}} \left\{ \frac{Pr(w_1^N) \cdot Pr(x_1^T|w_1^N)}{Pr(x_1^T)} \right\} \tag{1.12}$$

$$= \underset{N,w_1^N}{\operatorname{argmax}} \left\{ Pr(w_1^N) \cdot Pr(x_1^T|w_1^N) \right\} \tag{1.13}$$

The two resulting probabilities are usually referred to as the *language model* $Pr(w_1^N)$ which provides an a-priori probability for the word sequence w_1^N and the *acoustic model* $Pr(x_1^T|w_1^N)$ providing a posterior probability for the acoustic features x_1^T given a word sequence w_1^N. A typical ASR system architecture is presented in Figure 1.6.

In a first preprocessing step, the (analogue) audio signal is analyzed and converted into a sequence of feature vectors x_1^T, which are the input units for the actual statistical approach. For the global search or decoding of the acoustic vector sequence, the acoustic model and the language model have to be trained beforehand. The acoustic model is usually realized as a combination of models for the smallest (sub-) word units which should be distinguished by the speech recognition system. Depending on the task, these smallest units might be e.g. phonemes, subwords like syllables or even whole words. Additionally, a pronunciation lexicon is provided which defines "valid" sequences of subword units forming the words which can be produced by the acoustic model. The language model provides a probability for a hypothesized word sequence based on purely textual features like syntax, semantics, and pragmatics of the target language. Both models are combined and weighted within the search process to determine

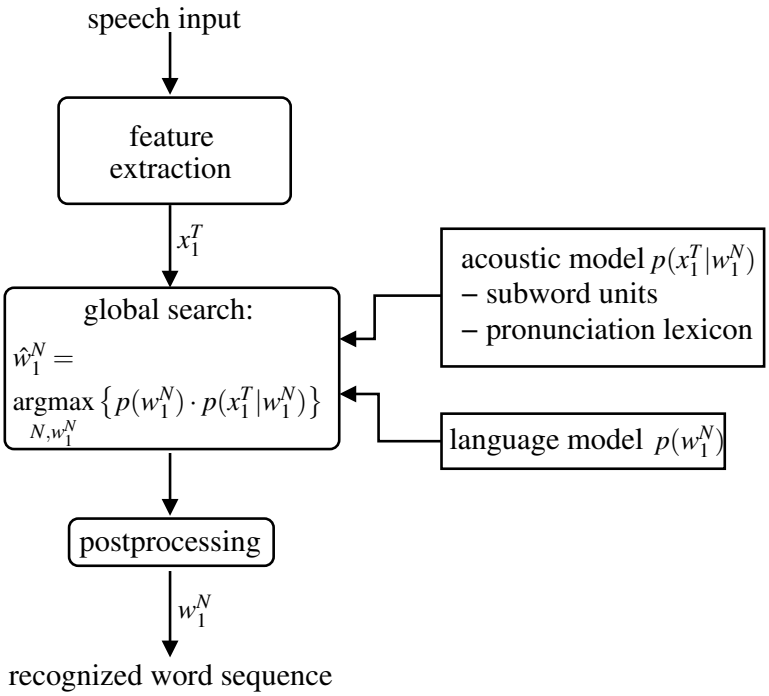


Figure 1.6 Diagram of a typical automatic speech recognition system following Bayes' decision rule [Ney 90]. The feature extraction provides the features which are used in the global search process together with the knowledge sources represented within the acoustic and language model.

the best mapping from the acoustic vectors to the word sequence $\hat{w}_1^N$. In a post-processing step, certain recognized units may be discarded (e.g. silence or noise markers) or modules for truecasing might be applied, if the pronunciation lexicon only contains lower-case words. In the following sections, the four main components of an ASR system are introduced in detail with pointers to literature.

1.3.2 Feature Extraction

The signal analysis module maps the acoustic signal to a sequence of acoustic vectors or observations. Since we are only interested in keeping the speech information and not the information about the speaker, a good feature extraction aims at reducing the feature vectors' dimensionality by omitting information about the speaker, the audio signal intensity, background noises, etc. while at the same time the characteristics of the (sub-) word units within the acoustic model should be kept so that those can be well discriminated.

Within a state-of-the-art LVCSR systems, a short term spectral analysis is the first step, usually a Fourier analysis [Rabiner & Schafer 79]. Techniques for further processing and smoothing are commonly applied. Noteworthy are the mel-frequency cepstral coefficients (MFCC) [Davis & Mermelstein 80], perceptual linear prediction (PLP) [Hermansky 90], and gammatone filter based features (GT) [Aertsen & Johannesma$^+$ 80] as well as their combina-

tion as e.g. presented in [Woodland & Gales+ 97, Schlüter & Bezrukov+ 07]. Common for all these different approaches is the idea to replicate the human auditory system. In recent years, methods to include phone posterior probability estimates have emerged. Here, features like MFCCs and PLPs are fed into a (hierarchical and/or recurrent) neural network which outputs the phone posteriors [Hermansky & Ellis+ 00, Valente & Vepa+ 07].

It is also desirable to include dynamic information within the feature vectors. Therefore, the original feature vector is augmented with the first and second order derivatives resulting in a feature vector with a very high dimensionality. To reduce the dimensionality, an approach based on linear discriminant analysis (LDA) is commonly used [Fisher 36, Duda & Hart+ 01]. Here, a linear transformation is applied which projects a feature space into a lower-dimensional subspace such that the class separability for distributions with equal variances are maximized. In a typical ASR system, 7–11 consecutive feature vectors are concatenated and the dimensionality is reduced to roughly 40.

Although there are steps integrated within the first signal analysis steps like MFCC or PLP extraction to reduce the speaker dependence, there is still a lot of speaker dependent information within the resulting features vectors. This has been demonstrated in several papers which use this kind of features to successfully detect gender [Stolcke & Bratt+ 00] or to even identify speakers [Doddington & Przybocki+ 00]. Thus, numerous methods have been developed to strive for a better speaker independence. Noteworthy are two commonly used approaches for speaker normalization and adaptation respectively, namely vocal tract length normalization (VTLN) and maximum likelihood linear regression (MLLR), which are usually used within a multi-pass ASR system. Within VTLN, a warping factor for a speaker is chosen empirically maximizing the likelihood of the speaker cluster given the recognition result of a former recognition pass [Acero 90, Wegmann & McAllaster+ 96]. Using a classifier to select the warping factor is more efficient and presented in [Molau 03]. Within the MLLR approach, the means and variances of the Gaussian mixture models (GMMs) are adapted to the speaker by applying a speaker-dependent linear transformation. There is also a constrained MLLR (C-MLLR), which uses the same matrix for the means and variances [Leggetter & Woodland 95]. Both variants are presented and compared in detail in [Pitz 05].

1.3.3 Acoustic Model

The acoustic model (AM) is one of the two main knowledge sources used in state-of-the-art ASR systems. It returns the likelihood of a feature vector x_1^T given a word sequence w_1^N. Using a so-called pronunciation dictionary, the sub-word units which are recognized by the ASR system are defined. In principle, it is possible to use whole words as recognition units. But this is only meaningful for tasks where the used vocabulary is very limited and constant, like e.g. digit recognition. Within LVCSR systems, usually phonemes respectively context-dependent phonemes are used. Thus, it is possible to reduce the model's complexity, since there are less phonemes in a language than whole words. Training of the model also improves, since there are usually more occurrences of the same phoneme within the training data than of a whole word. Thus, the problem of data sparsity is reduced. The AM is now actually a concatenation of AMs for these basic sub-word units. Additionally, the pronunciation dictionary defines the possible

words and thus the valid sequences of AMs. Since a finite pronunciation dictionary always leads to words which can not be recognized by the ASR system, so-called out of vocabulary (OOV) words, algorithms have been developed to recognize sub-word units directly and merge those to valid words within a post-processing step as e.g. presented in [Bisani & Ney 05, Basha Shaik & El-Desoky Mousa+ 11, El-Desoky Mousa & Basha Shaik+ 12]. Another approach to tackle the OOV problem is to include single characters as low-level recognition units, which is so far only working successfully for image recognition, where no pronunciations are necessary [Kozielski & Rybach+ 13].

Since the context information between phonemes seems to be crucial for good performance of LVCSR systems, so-called *n-phones* or *allophones* are usually used as basic recognition units. Most commonly, so-called triphones or quinphones are applied, which model the current phoneme in the context of one or two preceding and succeeding phonemes respectively. This phoneme context might also be modelled across word boundaries, which is called across-word modeling and also a standard method in current ASR systems [Hon & Lee 91, Odell & Valtchev+ 94, Sixtus 03].

Another complexity which has to be tackled by the AM is the high variability in speaking rate across different speakers and/or languages. Since the seventies, the de-facto standard approach are the so-called hidden Markov models (HMMs) [Baker 75, Rabiner & Juang 86]. An HMM is a stochastic finite state automaton (FSA) consisting of states and transitions connecting the states. These states are included in the original probability of the AM as a hidden variable and can be further broken down using Bayes' identity:

$$Pr(x_1^T|w_1^N) = \sum_{s_1^T:w_1^N} Pr(x_1^T, s_1^T|w_1^N) \tag{1.14}$$

$$= \sum_{s_1^T:w_1^N} \prod_{t=1}^{T} Pr(x_t|x_1^{t-1}, s_1^t; w_1^N) \cdot Pr(s_t|s_1^{t-1}; w_1^N) \tag{1.15}$$

Since the computational cost would be too high for this true probability, a first-order Markov model assumption is introduced [Duda & Hart+ 01]. Both probabilities are only conditioned on the immediate predecessor state:

$$p(x_1^T|w_1^N) = \sum_{s_1^T:w_1^N} \prod_{t=1}^{T} p(x_t|s_t, w_1^N) \cdot p(s_t|s_{t-1}, w_1^N) \tag{1.16}$$

The two resulting probabilities are usually referred to as *emission probability* $p(x_t|s_t, w_1^N)$ and *transition probability* $p(s_t|s_{t-1}, w_1^N)$. The sum over the states in Equation 1.16 is usually approximated by the maximum, which is often called *Viterbi* approximation [Ney 90]:

$$p(x_1^T|w_1^N) = \max_{s_1^T:w_1^N} \left\{ \prod_{t=1}^{T} p(x_t|s_t, w_1^N) \cdot p(s_t|s_{t-1}, w_1^N) \right\} \tag{1.17}$$

Both the probabilities within Equations 1.16 and 1.17 can be efficiently calculated using dynamic programming [Bellman 57, Viterbi 67, Ney 84], more precisely the forward-backward

algorithm is often used [Baum 72, Rabiner & Juang 86].

Until now, we did not yet define the structure of the HMM automaton. Usually, the so-called Bakis topology is used [Bakis 76]. Here, the basic HMM consists of six subsequent states, whereas the states one and two model the start of the phoneme, the states three and four the middle and the last two states the end of a phoneme. With respect to the modeled emission probabilities, the two states in each pair are identical. There exist three kinds of transitions between the states. Each state has a *loop* transition to the same state, a *forward* transition to the next state, and a *skip* transition to the next but one state. In relation to the "distance" covered by those transition, this topology is sometimes referred to as 0-1-2 model. Usually, a state represents a time frame of a length of 10ms. Thus, when only using forward transitions, the time to traverse an HMM is 60ms, which is roughly the average duration of a phoneme within most languages. Using the loop and skip transition, the time to traverse the 6-state HMM for a phoneme can be adopted to the speaking rate. For fast, conversational speech, the minimum of 30ms has been found to be too long [Molau 03]. Thus, the pairwise identical states are merged and a three state HMM is typically used. Now, it is possible to traverse the whole HMM in just 10ms to 20ms. To get a better idea of the actual structure, Figure 1.7 presents an example HMM for a part of the word "seven" using triphones as units per six state HMM.

Besides the choice for the HMM structure, the modeling of the emission and transition probability given in Equation 1.16 defines the AM. For the emission probability for the HMM states, usually Gaussian mixture models (GMMs) are used:

$$p(x|s;w_1^N) = \sum_{l=1}^{L_s} c_{sl}\,\mathcal{N}(x|\mu_{sl},\Sigma_{sl};w_1^N S) \tag{1.18}$$

Here, L_s Gaussian densities $\mathcal{N}(x|\mu_{sl},\Sigma_{sl};w_1^N S)$ are estimated per state, whereas the free parameters are the means μ_{sl}, the covariance matrices Σ_{sl} and the (non-negative) mixture weights c_{sl}. To obtain a proper GMM, the mixture weights per state have to sum up to one, i.e. $\sum_{l=1}^{L_s} c_{sl} = 1$. The emission probabilities are sometimes alternatively modeled using different distributions, e.g. discrete probabilities [Jelinek 76], semi-continuous probabilities [Huang & Jack 89] or other continuous distributions than Gaussian mixtures [Levinson & Rabiner$^+$ 83].

Within the RWTH Aachen University (RWTH) ASR system, which will be used for most of the experiments described in Chapter 9, a globally pooled and diagonal covariance matrix Σ is used instead of the full covariance matrix depicted in Equation 1.18. Thus, the set of free parameters is reduced to $\Lambda = \{\{\mu_{sl}\},\{c_{sl}\},\Sigma\}$. Typically, for a baseline ASR system, maximum likelihood (ML) training of these parameters using the expectation maximization (EM) algorithm is used [Dempster & Laird$^+$ 77].

Two more techniques are typically applied to tackle the problem of data sparseness as well as for efficiency reasons. On the one hand, especially when using a pooled and diagonal covariance matrix, decorrelated features are assumed, which is a side-effect of applying linear discriminant analysis (LDA) to include dynamic information as already presented in the previous feature extraction section [Fisher 36, Duda & Hart$^+$ 01]. Since the number of possible triphones or quinphones is usually too high to result in robust probability estimates per HMM state, n-phone or allophone, several states are tied together leading to generalized allophone

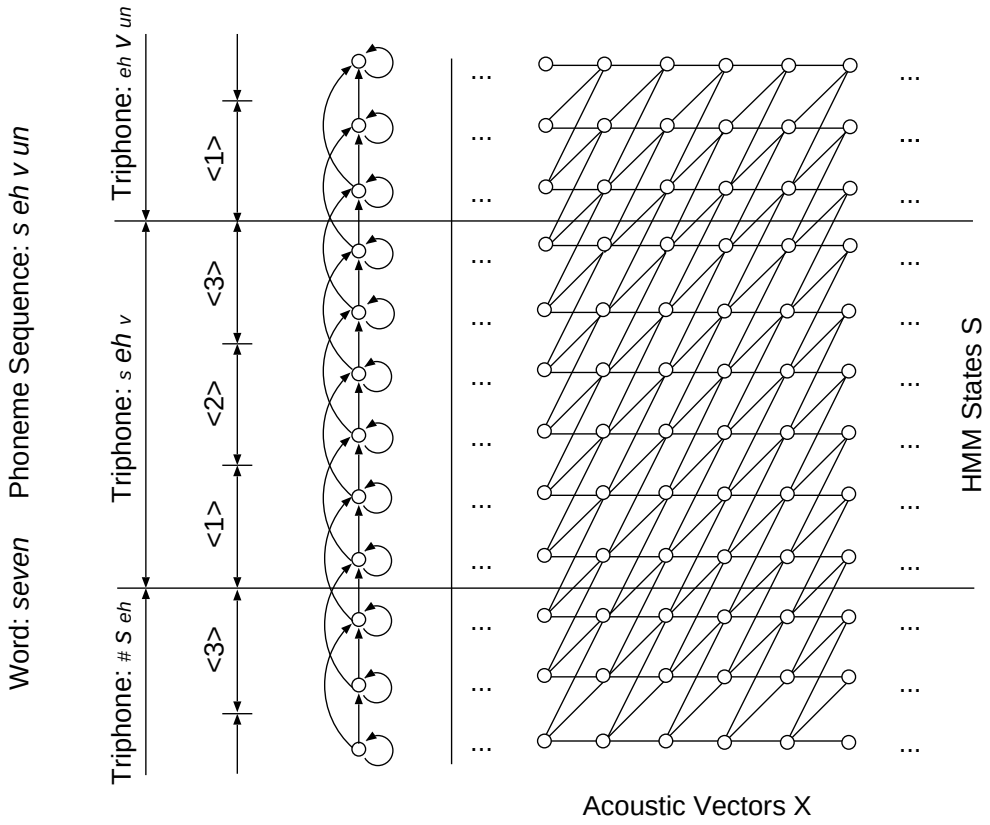


Figure 1.7 Example for a six state HMM in Bakis topology for the triphone $_{s}eh_{v}$ within the word "seven". The individual triphone begin, middle and end parts of the HMMs are marked with <1>,<2>, and <3>. Note that the HMMs modeling the triphones $_{\#}s_{e}$ and $_{eh}v_{un}$ are only partially visible.

models [Young 92]. Usually, state-of-the-art ASR systems apply a top-down state clustering based on decision trees, called classification and regression tree (CART) clustering. Using this approach, even allophones not seen in training are assigned to an appropriate HMM state without the need of a back-off model [Beulen 99].

1.3.4 Pronunciation Dictionary

The pronunciation dictionary is an important part of an ASR recognizer. Since one of the two main topics of this thesis is concerned with the G2P task, it deserves some special attention. For the design of an LVCSR system, usually two different pronunciation dictionaries are used, namely one for training of the AM and one for the recognition or decoding. The training lexicon usually contains all the words within the training data set and is necessary to obtain an alignment between the audio signal and the allophones. Here, the lexicon provides the mapping from the written word to the corresponding allophone sequence. The training data might contain broken words or other sounds like noises or breath which might be mapped to a "garbage collection" phoneme. For decoding, the selection of words is much more important

and depends on the task. Usually, large in-domain text sources are collected for language modeling cf. the following Section 1.3.5) and the vocabulary is selected using unigram count statistics. Named entities or words covering current events are often part of the recognition lexicon.

Since the quality of the phonetic transcription is crucial for training as well as for recognition, good G2P systems are needed, since the effort to transcribe the words manually is high with respect to both time and cost. Usually, there are background lexicons available which are used to train G2P systems and those are subsequently used to phonetically transcribe words which are not within these lexicons. More details on the need for phonemes as a mapping layer and the G2P system in general are given in Section 1.3.8.

1.3.5 Language Model

The language model (LM) provides an a-priori probability for a sequence of words, covering the syntax, semantics and pragmatics of the respective language implicitly. Due to the unlimited number of possible and valid sentences, some restrictions have to be applied to lead to robust probabilities. Within typical LVCSR systems, the so-called *m-gram* approach is used [Bahl & Jelinek$^+$ 83]. Here, the probability of the current word is conditioned only on a limited history of m preceding words:

$$Pr(w_1^N) = \prod_{n=1}^{N} Pr(w_n|w_1^{n-1}) \tag{1.19}$$

$$= \prod_{n=1}^{N} p(w_n|w_{n-m+1}^{n-1}) \tag{1.20}$$

$$= \prod_{n=1}^{N} p(w_n|h_n) \tag{1.21}$$

As shown in Equation 1.21, the word sequence w_{n-m+1}^{n-1} is denoted as *history* h_n of word w_n with length m. If $n<m$, i.e. within the first $m-1$ words of a sentence, the history is defined as $h := w_1^{n-1}$. At the sentence boundary, the history is empty. Note that depending on the task, sentence start and sentence end markers might be included within the LM. It is also possible to merge both to a sentence boundary marker.

The most frequently used training criterion for m-gram LMs is the co-called *log perplexity*:

$$\log PP(w_n) = \log \left[\prod_{n=1}^{N} p(w_n|w_{n-m+1}^{n-1}) \right]^{-1/N} \tag{1.22}$$

$$= -\frac{1}{N} \sum_{n=1}^{N} \log p(w_n|w_{n-m+1}^{n-1}) \tag{1.23}$$

Additionally, the perplexity is commonly used as a measure to evaluate the performance of LMs purely based on text data (i.e. not within an ASR system) [Brown & Pietra$^+$ 92]. It

can be interpreted as the average number of choices for a word to follow the word history h_n. Usually, the m-gram LM estimates are computed using relative frequencies on large corpora of transcriptions of speech and written text. Note that relative frequencies are the optimal solution for m-gram LMs if log perplexity is the training criterion, i.e. they minimize the log perplexity.

Since the number of possible m-grams increases exponentially with the history length m, a sparseness problem arises and a large number of m-grams will not be observed in training, assuming a large vocabulary. To tackle this issue, smoothing techniques have been introduced which subtract probability mass from seen events [Katz 87, Ney & Essen$^+$ 94, Generet & Ney$^+$ 95]. The "free" probability mass is either distributed over all unseen events, called *backing off* or over all events, referred to as *interpolation*, usually in combination with LMs with shorter history length. A comparison of smoothing techniques is presented in [Chen & Goodman 96]. To estimate the parameters of a smoothed LM, *leaving-one-out* is often applied [Ney & Martin$^+$ 97].

Within the ASR systems build for this work as well as for most systems build at RWTH, the SRI International (SRI) LM toolkit [Stolcke 02] has been applied for m-gram LM training, whereas mostly the so-called modified Kneser-Ney discounting with interpolation [Chen & Goodman 96] is applied as smoothing technique.

1.3.6 Search

Within the search process, the acoustic model and the language model are combined and used to find the word sequence $\hat{w}_1^N$ that maximizes the a posterior probability for a given sequence of acoustic feature vectors x_1^T (cp. Equation 1.13):

$$\hat{w}_1^N = \underset{N, w_1^N}{\operatorname{argmax}} \left\{ Pr(w_1^N | x_1^T) \right\} \tag{1.24}$$

$$= \underset{N, w_1^N}{\operatorname{argmax}} \left\{ Pr(w_1^N) \cdot Pr(x_1^T | w_1^N) \right\} \tag{1.25}$$

If we now substitute the general AM and LM probabilities in Equation 1.25 with the models derived in Equations 1.16 and 1.20, we get the actual optimization problem:

$$\hat{w}_1^N = \underset{w_1^N}{\operatorname{argmax}} \left\{ \left[\prod_{n=1}^{N} p(w_n | w_{n-m+1}^{n-1}) \right] \cdot \left[\max_{s_1^T} \prod_{t=1}^{T} p(x_t | s_t, w_1^N) \cdot p(s_t | s_{t-1}, w_1^N) \right] \right\} \tag{1.26}$$

Note that within the AM probability, the Viterbi approximation is already included which significantly reduces computational complexity. Usually, dynamic programming is used to solve this optimization problem [Bellman 57]. Here, due to properties of the mathematical structure, the problem is divided into sub-instances which can be solved individually.

There are basically two fundamental strategies to solve search problems, namely *depth-first* and *breadth-first* search. Within the former strategy, which is exploited by stack-decoding algorithms like Dijkstra [Dijkstra 59] or A^*-search [Jelinek 69, Paul 91], the state hypotheses are expanded time-*asynchronously* depending on a heuristic estimate of the costs to complete the

path. If the estimate is equal to the true cost, the search space is minimal. Within the breadth-first strategy, which is also used by the Viterbi search, the state hypotheses are expanded time-*synchronously* [Vintsyuk 71, Baker 75, Sakoe 79, Ney 84]. Here, the probabilities of all hypotheses are computed up to a certain time-frame and are thus comparable. Since it is only possible to simultaneously expand all hypotheses for small vocabularies W (the number of possible word sequences with maximum length N grows exponentially in N, more precisely W^N), pruning strategies have to be applied. In *beam-search*, only those hypotheses are expanded whose likelihood is sufficiently close to the likelihood of the current best hypothesis [Lowerre 76, Ney & Mergel$^+$ 87, Ortmanns & Ney 95]. Although by applying approximations like beam-search, the exact optimal solution for the search problem might not be found anymore, there are no significant search errors observed in practise, if the pruning parameters are adjusted properly.

Search complexity can be further reduced by applying one or several other techniques like e.g. lexical prefix trees [Ney & Häb-Umbach$^+$ 92], also in combination with LM look-ahead [Steinbiss & Ney$^+$ 93, Alleva & Huang$^+$ 96, Ortmanns & Ney$^+$ 96]. By restructuring search space [Ramasubramansian & Paliwal 92, Fritsch 97] or using the single instruction multiple data (SIMD) technique of modern CPUs [Kanthak & Schütz$^+$ 00], the computational time for decoding can be further reduced.

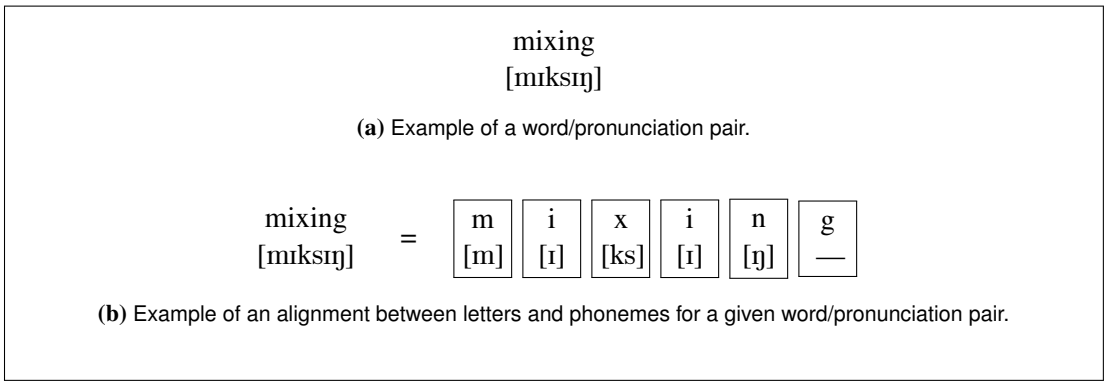
1.3.7 Further Techniques

Over the years, more advanced techniques have been introduced and are part of almost all state-of-the-art LVCSR systems. Concerning feature extraction, the use of neural networks (NNs), more precisely multi-layer perceptrons (MLPs), could improve ASR quality [Plahl & Kozielski$^+$ 13]. Recurrent neural networks (RNNs) could also be applied successfully for language modeling [Mikolov & Karafiát$^+$ 10, Mikolov & Deoras$^+$ 11]. With respect to the acoustic model, discriminative training techniques are usually applied, bootstrapped with a classical ML model [Woodland & Povey 02, Heigold 10].

Certain techniques can not be applied directly during a full search pass due to complexity reasons. Thus, it is common to use a *multipass* system. Here, in a first pass, parts of the search space are stored in either *n-best* lists [Schwartz & Chow 90, Schwartz & Austin 91] or *word graphs* [Ney & Oerder 93, Ney & Aubert 94, Ortmanns & Ney$^+$ 97]. Within a second recognition pass, this restricted search space is the basis to apply larger m-gram LMs, the so-called LM re-scoring, or even more complex acoustic models.

1.3.8 Grapheme-to-Phoneme Conversion

Besides the concept tagging task within NLU systems, building and evaluating grapheme-to-phoneme conversion (G2P) systems is the second challenge covered in this thesis. Within this task, sequences of graphemes have to be translated into corresponding sequences of phonemes. Here, a *grapheme* is defined as a symbol used for writing language (e.g. a letter) whereas a phoneme is considered to be the smallest contrastive unit in the sound system of a language. More formally, given an orthographic form (grapheme sequence) $\mathbf{g} \in G^*$, the task is to find the most likely pronunciation (phoneme sequence) $\boldsymbol{\varphi} \in \Phi^*$:



(a) Example of a word/pronunciation pair.

(b) Example of an alignment between letters and phonemes for a given word/pronunciation pair.

Figure 1.8 Example for a word/pronunciation pair and the corresponding alignment between letters and phonemes (graphones).

$$\boldsymbol{\varphi}(\mathbf{g}) = \underset{\boldsymbol{\varphi}' \in \Phi^*}{\operatorname{argmax}}\, p(\mathbf{g}, \boldsymbol{\varphi}') \tag{1.27}$$

An example for a word/pronunciation pair is given in Figure 1.8a.

Here, the square brackets on phoneme side are part of the Speech Assessment Methods Phonetic Alphabet (SAMPA) notation and not of the phonemes. These brackets denote a phonetic transcript in contrast to regular text. In contrast to the concept tagging task, there is usually no alignment between graphemes and phonemes provided within the training data. Thus, an additional level of complexity is introduced. Many methods make the assumption that for each word, its orthographic form and its pronunciation are generated by a common sequence of blocks that carry both, letters and phonemes. In the literature, such a block is called grapheme-phoneme, joint-multigram, or *graphone* for short [Bisani & Ney 08]. Formally, a graphone is a pair of a letter sequence and a phoneme sequence of possibly different length:

$$q = (\mathbf{g}, \boldsymbol{\varphi}) \in Q \subseteq G^* \times \Phi^* \tag{1.28}$$

An example for a graphone sequence is given in Figure 1.8b. Depending on the phoneme set, graphones might be asymmetric or even an empty grapheme/phoneme might be needed to form a correct alignment. While widely used in G2P approaches, the original idea of using graphones has been introduced in [Deligne & Yvon$^+$ 95]. Within log-linear methods, like CRFs, the alignment is usually integrated as a hidden variable (cf. Chapter 8).

More detailed information about the G2P task as well as a description of the various models and tasks considered in this work is given in Chapter 6.

1.3.8.1 Why is ASR using Phonemes?

Since G2P models are mostly used within LVCSR systems, the need for using phonemes and thus also such models should be investigated. Although phonemes describe the sounds of a language, the purpose of using phonemes in ASR is *state tying*. A phoneme is just a label for

an HMM as introduced in Section 1.3.3, or more typically a predicate in the state tying decision tree. It has been shown that the letters of the written form (graphemes) can be used directly and the rest is left to the acoustic modeling [Kanthak & Ney 03]. This works well for words that have a normal or regular pronunciation or that are very frequent. But for the long tail of infrequent words and/or those which have strange pronunciations, it does not work well. Thus, for a practical system, there is a need for a mapping layer between written and spoken forms to cope with irregularities in this relation. A nice example of words which are written similarly but have very different pronunciation would be "Winchester" [wɪntʃɪstə(r)] versus "Worchester" [wʊstə(r)]. Also, there are named entities with unusual character sequences, e.g. the artist "Ke$ha". Linguists know how to describe pronunciations using phonemes (more often than not) consistently and unambiguously. But since the manual transcription of pronunciations by linguistic experts is costly with respect to time and money, G2P systems are still an important part of any LVCSR system.

1.3.8.2 Why not just using a Dictionary?

Another argument against the need of G2P systems might be that there are plenty and large pronunciation lexica available, which have been manually verified. This might work for small vocabulary systems (e.g. digit recognition), where individual acoustic models for each word are trained. Large vocabulary systems operate with a fixed large but finite vocabulary, usually containing 10k-100k word forms. Again, for a dictation task within a fixed and limited domain, this might still work, but this approach is less suitable for open vocabulary settings like broadcast news or podcasts. The number of different words does not seem to be finite and important content words change over time. Additionally, not all words are known beforehand. Thus grapheme-to-phoneme conversion is needed to generalize beyond a fixed set of words, especially for applications where end-user customization of the vocabulary is used.

1.4 Log-Linear Models

One focus in this work is the use of log-linear or discriminative models to solve monotone string-to-string translation tasks. A good overview about the various approaches is given in [Heigold 10].

For the work at hand, we are using two log-linear models, which only differ in the normalization term. The first one is normalized on a positional level (maximum entropy Markov model (MEMM) [McCallum & Freitag$^+$ 00]) and the second one on sentence level (conditional random field (CRF) [Lafferty & McCallum$^+$ 01]). For the sake of simplicity, we will use the notation from the concept tagging task as introduced in Section 1.2.1 instead of introducing an additional, more general notation. The general representation of these models is described in Equation 1.29 as a conditional probability of a concept tag sequence $t_1^N = t_1, \ldots, t_N$ given a word sequence $w_1^N = w_1, \ldots, w_N$:

$$p_{\lambda_1^M}(t_1^N | w_1^N) = \frac{1}{Z} \prod_{n=1}^{N} \exp\left(\sum_{m=1}^{M} \lambda_m \cdot h_m(t_{n-1}, t_n, w_1^N) \right) \tag{1.29}$$

...	$\boxed{w_{n-2}}$	w_{n-1}	w_n	w_{n+1}	w_{n+2}	...
...	t_{n-2}	t_{n-1}	$\boxed{t_n}$	x	x	...

Figure 1.9 Example for a lexical feature.

...	w_{n-2}	w_{n-1}	w_n	w_{n+1}	w_{n+2}	...
...	t_{n-2}	$\boxed{t_{n-1}}$	$\boxed{t_n}$	x	x	...

Figure 1.10 Example for a bigram feature.

The log-linear models are based on feature functions $h_m(t_{n-1}, t_n, w_1^N)$ representing the information extracted from the given utterance, the corresponding parameters λ_m which are estimated in a training process, and a normalization term Z. This normalization term will be introduced and discussed for both models considered separately in Sections 1.4.2 and 1.4.3.

1.4.1 Feature Functions

In our experiments we use binary feature functions h_m. If a pre-defined combination of the values $t_{n-1}, t_n, w_1, \ldots, w_N$ is found in the data, the value "1" is returned, otherwise the value is "0". For instance a feature function may fire if and only if

- the predecessor word w_{n-1} is "the" and the concept tag t_n is "name"
- the predecessor concept tag t_{n-1} is "number" and the concept tag t_n is "currency"
- the prefix (resp. word stem) of a word $w_n =$ "euros" of length $\delta = 4$ is "euro" and the concept tag t_n is "currency"

We will call the feature functions based on the current symbol on target side and any source symbol in a certain distance (e.g. predecessor, current, and successor words) *lexical features* and the features based on the predecessor concept *bigram features*. Features based on word parts (i.e. prefixes, suffixes, capitalization) are referred to as *word part features*. While the word part features are only used for the concept tagging task, where actual words are used as input, lexical and bigram features are task independent. Examples for the latter features are given in Figures 1.9 and 1.10.

If not stated otherwise, feature cut-offs are not applied. Thus, in general, a feature h_m is used if it is seen with any combination of t_n, t_{n-1}, and w_1^N in the training corpus. For clarity, we will abbreviate the term in the numerator of Equation 1.29 by:

$$H(t_{n-1}, t_n, w_1^N) = \exp\left(\sum_{m=1}^{M} \lambda_m \cdot h_m(t_{n-1}, t_n, w_1^N)\right) \tag{1.30}$$

resulting in

$$p_{\lambda_1^M}(t_1^N|w_1^N) = \frac{1}{Z}\prod_{n=1}^{N} H(t_{n-1}, t_n, w_1^N) \tag{1.31}$$

1.4.2 Maximum Entropy Markov Models (MEMM)

A possible normalization of Equation 1.31 is on a positional level:

$$Z = \prod_{n=1}^{N}\sum_{\tilde{t}_n} H(t_{n-1}, \tilde{t}_n, w_1^N) \tag{1.32}$$

resulting in the MEMM:

$$p_{\lambda_1^M}(t_1^N|w_1^N) = \frac{1}{\prod_{n=1}^{N}\sum_{\tilde{t}_n} H(t_{n-1}, \tilde{t}_n, w_1^N)}\prod_{n=1}^{N} H(t_{n-1}, t_n, w_1^N) \tag{1.33}$$

$$= \prod_{n=1}^{N}\frac{H(t_{n-1}, t_n, w_1^N)}{\sum_{\tilde{t}_n} H(t_{n-1}, \tilde{t}_n, w_1^N)} \tag{1.34}$$

$$= \prod_{n=1}^{N}\frac{\exp\left(\sum_{m=1}^{M}\lambda_m \cdot h_m(t_{n-1}, t_n, w_1^N)\right)}{\sum_{\tilde{t}_n}\exp\left(\sum_{m=1}^{M}\lambda_m \cdot h_m(t_{n-1}, \tilde{t}_n, w_1^N)\right)} \tag{1.35}$$

Here, $\tilde{t}_n$ stands for all possible concept tags. This modeling approach is also referred to as maximum entropy Markov models (MEMMs) [McCallum & Freitag+ 00], maximum entropy approach [Bender & Macherey+ 03], or log-linear on position level [Hahn & Lehnen+ 08a] in the literature.

1.4.2.1 Training

Using Equation 1.35 and a given training dataset $\{\{\bar{t}_1^N\}_s, \{w_1^N\}_s\}_{s=1}^S$, the training of such a MEMM can be performed using the maximum class posterior probability as training criterion:

$$\hat{\lambda}_1^M = \underset{\lambda_1^M}{\operatorname{argmax}}\left\{\sum_{s=1}^{S}\log p_{\lambda_1^M}(\{\{\bar{t}_1^N\}_s, \{w_1^N\}_s\})\right\} \tag{1.36}$$

Here, $\bar{t}_1^N$ denotes the reference concept tag sequence. This training criterion is convex (i.e. there is only a single global maximum). Resilient backpropagation (RProp) is used to iteratively find the optimal λ_1^M. For the actual training process, Gaussian priors are assumed on the maximum entropy parameters for smoothing. This is equivalent to using a L2 regularization weighted by a constant c:

$$\hat{\lambda}_1^M = \underset{\lambda_1^M}{\operatorname{argmax}}\left\{\sum_{s=1}^{S}\log p_{\lambda_1^M}(\{\bar{t}_1^N\}_s|\{w_1^N\}_s) + \log p(\lambda_1^M)\right\} \tag{1.37}$$

$$= \underset{\lambda_1^M}{\operatorname{argmax}}\left\{\sum_{s=1}^{S}\log p_{\lambda_1^M}(\{\bar{t}_1^N\}_s|\{w_1^N\}_s) - c\,||\lambda_1^M||^2\right\}, \tag{1.38}$$

where

$$p(\lambda_1^M) = \prod_{m=1}^{M} \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{\lambda_m^2}{2\sigma^2}\right] \tag{1.39}$$

and

$$||\lambda_1^M||^2 = \sum_{m=1}^{M} \lambda_m^2 \tag{1.40}$$

Assuming a zero-mean Gaussian prior on the λ_1^M with independence across dimensions and a single tied variance term as in Equation 1.39, the equivalence of using a Gaussian prior and L2 regularization in the training criterion can be derived as follows:

$$\log p(\lambda_1^M) = \sum_{m=1}^{M} \log(\frac{1}{\sqrt{2\pi}\sigma}) - \frac{\lambda_m^2}{2\sigma^2} \tag{1.41}$$

$$= Const(\lambda) - \frac{1}{2\sigma^2} \sum_{m=1}^{M} \lambda_m^2 \tag{1.42}$$

$$= Const(\lambda) - c\,||\lambda_1^M||^2 \tag{1.43}$$

Here, the constant term depending on λ can be dropped when used within Equation 1.37 due to the argmax operation resulting in Equation 1.38.

1.4.2.2 Search

The best concept tag sequence $\hat{t}_1^N$ is derived by maximizing $p_{\lambda_1^M}(t_1^N|w_1^N)$ over the concept tag sequence:

$$w_1^N \rightarrow \hat{t}_1^N = \underset{t_1^N}{\text{argmax}}\, Pr(t_1^N|w_1^N) \tag{1.44}$$

$$= \underset{t_1^N}{\text{argmax}} \left\{ p_{\lambda_1^M}(t_1^N|w_1^N) \right\} \tag{1.45}$$

$$= \underset{t_1^N}{\text{argmax}} \left\{ \prod_{n=1}^{N} \frac{\exp\left(\sum_{m=1}^{M} \lambda_m \cdot h_m(t_{n-1}, t_n, w_1^N)\right)}{\sum_{\tilde{t}_n} \exp\left(\sum_{m=1}^{M} \lambda_m \cdot h_m(t_{n-1}, \tilde{t}_n, w_1^N)\right)} \right\} \tag{1.46}$$

1.4.3 Linear Chain Conditional Random Fields (CRF)

Linear chain conditional random fields (CRFs) as defined in [Lafferty & McCallum+ 01] can be represented within the same mathematical framework as the MEMMs, i.e. Equation 1.29. The only difference is the normalization term, which is now on *sentence* level:

$$Z = \sum_{\tilde{t}_1^N} \prod_{n=1}^{N} H(\tilde{t}_{n-1}, \tilde{t}_n, w_1^N) \tag{1.47}$$

This leads to the following model representation:

$$p_{\lambda_1^M}(t_1^N|w_1^N) = \frac{1}{\sum_{\tilde{t}_1^N}\prod_{n=1}^N H(\tilde{t}_{n-1},\tilde{t}_n,w_1^N)}\prod_{n=1}^N H(t_{n-1},t_n,w_1^N) \tag{1.48}$$

$$= \frac{\prod_{n=1}^N H(t_{n-1},t_n,w_1^N)}{\sum_{\tilde{t}_1^N}\prod_{n=1}^N H(\tilde{t}_{n-1},\tilde{t}_n,w_1^N)} \tag{1.49}$$

$$= \frac{\prod_{n=1}^N \exp\left(\sum_{m=1}^M \lambda_m \cdot h_m(t_{n-1},t_n,w_1^N)\right)}{\sum_{\tilde{t}_1^N}\prod_{n=1}^N \exp\left(\sum_{m=1}^M \lambda_m \cdot h_m(\tilde{t}_{n-1},\tilde{t}_n,w_1^N)\right)} \tag{1.50}$$

Again, $\tilde{t}_1^N$ represents all possible concept tag sequences. For CRFs, the same training and decision criteria as for MEMMs are used (cf. Equations 1.37 & 1.45). Note that CRFs are also convex, but due to the interchanged sum and product in the denominator, they are much more computationally expensive.

In [Heigold & Schlüter+ 09], the idea of merging the optimization of feature weights (training) based on SVMs and CRFs, called maximum mutual information (MMI) there, is described. The authors start from an SVM training process described by

$$\hat{\lambda}_1^M = \underset{\lambda_1^M}{\operatorname{argmax}}\left\{-\frac{1}{S}\sum_{s=1}^S l(\{\bar{t}_1^N\}_s,d,\rho) - c||\lambda_1^M||^2\right\} \tag{1.51}$$

with the distance

$$d = \sum_{m=1}^M \lambda_m\left(h_m(\{\bar{t}_1^N\}_s,w_1^N) - h_m(t_1^N,w_1^N)\right) \tag{1.52}$$

and the hinge loss

$$l(\bar{t}_1^N,d,\rho) = \max_{t_1^N\neq\bar{t}_1^N}\left\{\max\{\rho - d(\bar{t}_1^N,t_1^N),0\}\right\} \tag{1.53}$$

Equation 1.36 and 1.51 differ mainly in the loss function. They smoothed the loss function, used the accuracy instead of the 0/1 loss, and added it to the loss function of MMI resulting in a modified posterior defined as:

$$p_{\Lambda,\rho}(t_1^N|w_1^N) = \frac{1}{Z'}\exp\left(\sum_{i=1}^I \lambda_i f_i(t_1^N,w_1^N) - \rho\mathscr{A}(t_1^N,\bar{t}_1^N)\right) \tag{1.54}$$

The normalization constant Z' is similarly defined as above:

$$Z' = \sum_{\tilde{t}_1^N}\exp\left(\sum_{i=1}^I \lambda_i f_i(\tilde{t}_1^N,w_1^N) - \rho\mathscr{A}(\tilde{t}_1^N,\bar{t}_1^N)\right) \tag{1.55}$$

Here, the margin score is set to the word accuracy

$$\mathscr{A}(t_1^N,\bar{t}_1^N) = \sum_{n=1}^N \delta(t_n,\bar{t}_n) \tag{1.56}$$

between the hypothesis t_1^N and the reference $\bar{t}_1^N$, scaled by $\rho \geq 0$. The margin-based training criteria are obtained by replacing the posterior. Note that only the training and not the decision process is changed. Further extensions of CRFs have been proposed, e.g. triangular-CRFs within the SLU context, taking dialog manager states into account [Jeong & Geunbae Lee 08].

If not stated otherwise, the presented results using the CRF approach always include the margin term. A detailed comparison of CRFs with and without margin term can be found in Section 4.3, Chapter 5 and [Hahn & Lehnen$^+$ 09].

1.5 Structure of this Document

The remainder of this work is structured as follows. Chapter 2 gives an overview of the scientific goals of this thesis, while in Chapter 3, a short wrap-up of related work and pointers to literature are given. In Chapter 4, the work related to NLU, i.e. concept tagging, is presented, extending the general theoretical background presented in this chapter as well as presenting experimental results for numerous methods to tackle this task. Chapters 6–9 are dealing with the second large part of this work, which is grapheme-to-phoneme conversion. While Chapter 6 presents a comparison of various generative strategies, Chapter 7 introduces CRFs as a discriminative method. CRFs are extended to HCRF including an implicit alignment in Chapter 8. In the following Chapter 9, G2P results using CRFs within LVCSR systems are presented and compared to a generative system. This work concludes in Chapter 10 with a summary and detailing the scientific contributions. Chapter 11 discusses some possible further research ideas.

Within the appendix A, the corpora and systems used are presented. Following are lists of figures, tables and symbols for a quick overview. A glossary and an acronym section are given to enhance the reading flow, while a comprehensive bibliography is given for reference.

Chapter 2

Scientific Goals

Numerous methods to tackle (monotone) translation tasks are proposed in the literature, but there are no detailed comparisons available for the typically applied methods. Within this thesis, we take a closer look at two monotone string-to-string translation tasks, namely *concept tagging* and *grapheme-to-phoneme conversion*. The main difference between these two tasks is the need to (explicitly) model an alignment between the source and target side. Whereas for concept tagging, there is usually an alignment given between words and concepts within the training data, for G2P, an alignment between graphemes and phonemes has to be derived automatically. Recently, linear chain conditional random fields (CRFs) have been introduced [Lafferty & McCallum$^+$ 01]. This discriminative log-linear modeling approach is well suited for monotone translation problems.

The objective of this thesis is to establish extensive comparisons between state-of-the-art methods for both tasks for various languages as well as to improve these results by applying and tuning CRFs. The theoretical and experimental goals of this thesis include:

A comparison of state-of-the-art methods for concept tagging

To establish baselines for the concept tagging task, various well-known and state-of-the-art approaches are applied to and evaluated on the same data sets. This is always done on several languages and corpora to get an idea of the robustness and multi-lingual quality of the methods considered. The comparisons always include training of the statistical models, tuning of the parameters on a development set, and the final performance measurement on an evaluation set containing unseen data. The respective work is presented in Chapter 4. Here, results are presented for both, attribute name and attribute value extraction as well as for two input modalities (manual transcriptions and speech input). French, Polish and Italian tasks of varying complexity are considered.

A comparison of state-of-the-art methods for grapheme-to-phoneme conversion

As for concept tagging, it is important to establish baselines. Thus, various well-known and state-of-the-art methods are compared with each other on the same data sets. Since G2P performance varies depending on the language, several languages and corpora are considered, with varying size and complexity. The comparisons always include training of statistical models, tuning of the parameters on a development set, and the final performance measurement on an evaluation set containing unseen data. For G2P, the respective findings are presented in Chapter 6. Here, besides a freely available medium-sized English task, large pronunciation

dictionaries in English, German, French, Italian and Dutch are explored.

Application of ROVER system combination

Additionally to applying single methods to both tasks, we are also interested in getting an idea of the complementarity of the approaches. Therefore, recognizer output voting error reduction (ROVER) system combination is applied including all systems for concept tagging as well as G2P. Besides measuring the effect on the error rate, an extensive discussion and error analysis is especially interesting for feature-based approaches, which could be extended with additional features derived from the insights of the system combination experiments. The respective results are presented in Chapter 4 for the concept tagging tasks and in Chapter 6 for the G2P task.

Application of CRFs to concept tagging

A CRF framework has been developed as a module to the RWTH Aachen University Speech Recognition (RWTH ASR) engine. With this in-house CRF realization, all the reported experiments have been performed. Besides a comparison to state-of-the-art methods, the selection and tuning of features as well as the feature build-up are presented (cf. Chapter 4). Additional to the typical rule-based approach to attribute name extraction, a stochastic two-level approach based on CRFs is derived and combined with the rule-based approach (cf. Chapter 4.4). Additionally, the standard CRF training criterion is extended by a margin term (cf. Chapter 5).

Application of (H)CRFs to grapheme-to-phoneme conversion

The G2P task has different requirements w.r.t. the features functions, which have to be modified accordingly. Due to the much larger number of features, methods to filter respectively reduce the number of active features have to be derived (cf. Chapter 7). Most importantly, there is no alignment between graphemes and phonemes provided with the training data. Thus, methods have to be explored to provide such an alignment for the CRF model training. Various methods are tested leading to the integration of the alignment as a hidden variable (hidden conditional random fields (HCRFs)). The respective findings are presented and compared in Chapter 8.

Investigations on the effect of using HCRF G2P within an LVCSR system

Although the phoneme error rates (PERs) for typical G2P systems are already pretty low (<10%), there might be an additional effect on LVCSR results when switching from a standard, state-of-the-art G2P system based on joint-n-grams towards a CRF-based G2P system. Additionally, the number of pronunciation variants as well as the kind of pronunciation score might influence the results. The respective findings are presented in Chapter 9.

Chapter 3

Related Work

In this chapter, an overview of literature related to the core methods presented in this thesis is given. This includes pointers which do not fit into any of the following chapters. More references w.r.t. the tackled tasks and the historical development are given in Chapter 1. A more detailed discussion of the specified approaches is given in the respective chapters.

3.1 NLU - Concept Tagging

Most of the work on concept tagging for NLU presented in this thesis, i.e. a comparison of various methods, has been published in [Hahn & Dinarelli$^+$ 11]. In the latter publication, FST, DBN, SMT, SVM, MEMM and CRF have been compared on several tasks, whereas the CRF approach outperformed all other tested approaches. Within [Dinarelli & Moschitti$^+$ 12], the authors applied a discriminative re-ranking scheme to the concept tagging task. An SVM system has been used to re-rank an n-best list provided by an FST approach. Especially when speech is used as input for concept tagging, a joint decoding of ASR and semantic tagging might be considered, as presented in [Deoras & Sarikaya$^+$ 12]. The more general question of the possibility of further improving part-of-speech (POS) tagging without an improved linguistic foundation is discussed in [Manning 11]. A comprehensible overview of the SLU task in general as well as the technical background for each of the modules is given in the book by Gokhan Tur and Renato De Mori [Tur & De Mori 11]. For a summary of well established SLU tasks and corresponding benchmark data resources, the reader is referred to [Tur & Wang$^+$ 13].

3.2 ASR - Grapheme-to-Phoneme Conversion

Although there are some rule-based approaches published as e.g. in [Pagel & Lenzo$^+$ 98], many of the presented methods related to the G2P task are based on a statistical joint-n-gram approach relying on so-called graphones as units which has been introduced in [Deligne & Yvon$^+$ 95]. Since then, many statistical approaches for tackling the G2P task have been published.

The author of [Kneser 00] (not publicly available) used a combination of decision trees and n-gram model. The work presented in [Galescu & Allen 02] presents an early investigation of applying joint-n-grams to tackle the G2P problem. In the following years, this approach has become more popular, e.g. the authors of [Chen 03] and [Vozila & Adams$^+$ 03] relied on graphone-based joint maximum entropy (ME) n-gram models. In [Bisani & Ney 02] and

[Bisani & Ney 08], a graphone-based ML-trained ME n-gram model has been proposed, which does still lead to state-of-the-art results and is available as open source tool [Bisani 08].

Recently, discriminative techniques have been applied to the G2P task, e.g. the work presented in [Jiampojamarn & Kondrak 09] and [Jiampojamarn & Cherry$^+$ 10] based on an online discriminative training framework in combination with a phrase-based SMT decoder. The results are very good, but the method is computationally expensive.

Another interesting approach is presented by the authors of [Novak 11]. Here, weighted FST-based decoding using n-gram LMs and an advanced M-to-N alignment algorithm is applied resulting in a very fast decoder with competitive results. This work has been recently extended by recurrent neural networks and is also available as open source [Novak & Dixon$^+$ 12]. First applications of weighted FSTs to the G2P problem are presented in [Caseiro & Trancoso$^+$ 02].

Additionally to the CRF approach to G2P conversion described in this work, the authors of [Wang & King 11] also applied CRFs to this task successfully.

Another interesting topic is how to measure the quality of a G2P system. Here, some ideas go beyond measuring PER and word error rate (WER) and try to weight different errors based on phoneme similarity metrics [Hixon & Schneider$^+$ 11].

Chapter 4

Methods for Concept Tagging - A Comparison

Within a dialogue system, the extraction of flat concepts out of a given word sequence (usually provided by an ASR system) is one of the first steps of any spoken language understanding (SLU) system. A detailed description of the various modules of an NLU system is given in Section 1.2, whereas related work is presented in Section 3.1. These concept tags do basically segment the input sentence into chunks, whereas there is always a tag to label chunks without semantic meaning for the specific task. One example from the ATIS corpus, a very early data collection concerned with negotiating air travel with a travel planer, is given in Figure 4.1. Here, the annotation of the spoken sentence with concept tags is divided into two levels: the attribute name (annotated in the figure in bold to the left of the transcription) and the corresponding attribute value (annotated in the figure in square brackets to the right of the transcription). A more detailed and recent study concerned with this corpus can be found in [Tur & Hakkani-Tur$^+$ 10]

Within this chapter, six different modeling approaches are presented and compared to tackle the task of concept tagging. The methods include well-known generative and discriminative approaches like finite state transducers (FSTs), phrase-based statistical machine translation (SMT), support vector machines (SVMs) or maximum entropy Markov models (MEMMs). Additionally, approaches only recently applied to natural language processing like conditional random fields (CRFs) and dynamic Bayesian networks (DBNs) are considered. Except for the CRF and the closely related MEMM model, which have been introduced in great detail in Section 1.4, a detailed description of the models is presented.

After some remarks concerning the alignment between the word sequence and the attribute names, which is needed for most of the presented models for training, experimental and comparative results for theses models are presented on three corpora in different languages and with different complexity. The French MEDIA corpus has already been exploited during an evaluation campaign and so a direct comparison with existing benchmarks is possible [Bonneau-Maynard & Ayache$^+$ 06]. Two more corpora have been collected within the EU FP6 Spoken Languange UNderstanding in MultilinguAl Communication Systems (LUNA) project: the Polish Warsaw transportation corpus [Mykowiecka & Marasek$^+$ 09] and the Italian help-desk corpus [Dinarelli & Quarteroni$^+$ 09]. A detailed description of these three corpora is given in Section A.1.

The considered corpora have ontologies of different types and complexity that can be represented in a frame language described in [De Mori & Hakkani-Tur$^+$ 08]. In tasks like MEDIA, there are frames describing properties of objects in application domains and frames describing dialog acts. These frames have some properties whose values are instances of other frames

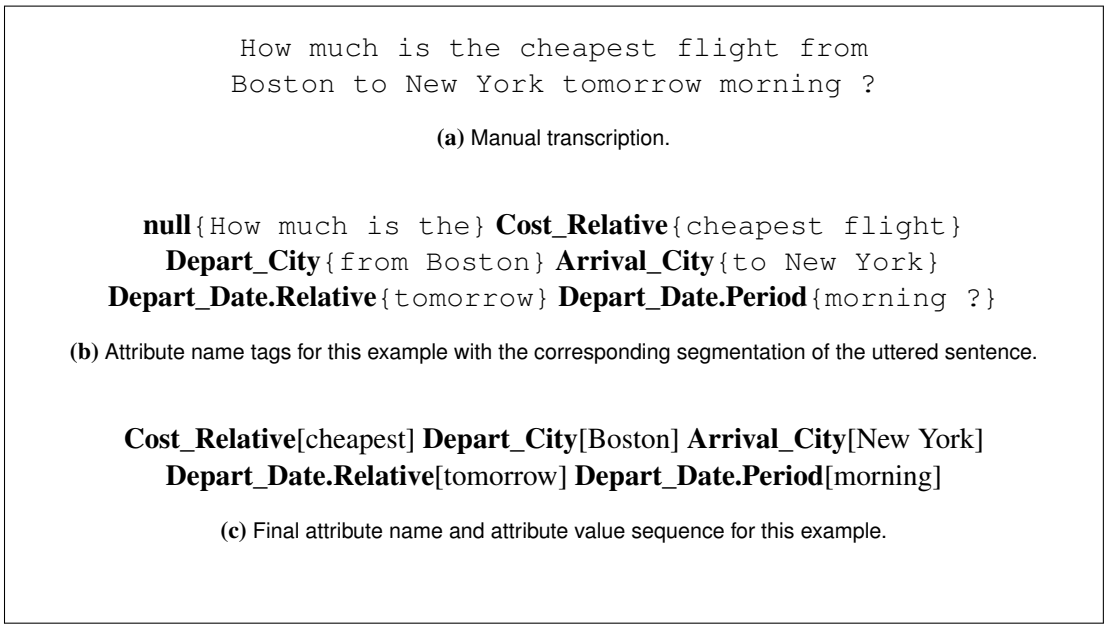


Figure 4.1 Example sentence from the English ATIS corpus illustrating concept tagging with the two levels attribute name and attribute value.

resulting in fairly complex semantic structures. Attribute value logical predicates can be obtained from these frames, an attribute being a frame property. When the value of a property is a frame structure, this structure can be represented by a semantic class name. For example, the request for a reservation is represented by a frame REQUEST that has a property with name `request_object`. Value types for an object representing this property are listed in the slot facet of the property. The facet of `request_object` contains a structure type represented by a semantic class whose name is RESERVATION. In the MEDIA annotation a name corresponding to the property `request_object` of the frame REQUEST will have values corresponding to the elements of the property slot facet. References are also examples of other elements in the facet. In case of ambiguities, disambiguation is performed by constituent composition, a process that is not described in this work.

A distinction is made between two tasks: extraction of only attribute names and extraction of attribute names with corresponding attribute values. Whereas the first task requires solely a segmentation and tagging of the input sentence, the second task additionally requires the extraction of a (normalized) value out of the given word sequence chunk together with the corresponding concept tag.

Additionally, two conditions are considered, namely manual transcriptions of word hypotheses as input, which can be considered more or less flawless, and automatically generated transcriptions using an ASR system. While the manual transcriptions are used to analyze the potential of the various models, the error prone ASR transcriptions are necessary to analyze the robustness as well as the usability in real-life dialog systems. Additionally to single systems,

methods for system combination are also considered. Here, the focus is on using ROVER for all six methods.

Experiments reported in this thesis show that CRFs systematically outperform all the other methods even using fairly simple functions in the model exponents. The proposed CRFs seem to model the overall expression of a concept better than the other considered models when this semantic information is conveyed by word sequences. This does not appear to be the case for spoken opinion analysis performed on arbitrarily long telephone messages and dialogs as described in [Camelin & Béchet$^+$ 10].

There are certain similarities between tasks such as part-of-speech (POS) tagging [Schmid 94], name transliteration [Deselaers & Hasan$^+$ 09] or grapheme-to-phoneme conversion [Jiampojamarn & Kondrak 09] and concept tagging suggesting that some findings described in this work may be helpful also for these tasks. Note that for the task of concept tagging, usually an alignment between the input sentence and the concept tags (i.e. a segmentation of the data) is provided with the training data. This is not the case for tasks like grapheme-to-phoneme conversion or transliteration, which does introduce another degree of complexity for those tasks.

Since the performance of CRFs and the comparison to other methods is a scientific goal of this work, this chapter does focus on analyzing this approach. Most of the findings in this chapter have been published in [Hahn & Dinarelli$^+$ 11]. In addition to the extensive and consistent experimental comparison of six different statistical methods on three corpora in different languages, this chapter presents improved CRFs by introducing margin posteriors leading to best published results on the MEDIA corpus in relaxed-simplified condition, ROVER system combination using all six systems all carefully tuned on exactly the same data and statistical improved attribute value extraction using CRFs in combination with rule-based attribute value extraction. The improved CRF training criterion is presented in detail in Chapter 5. Parts of the presented work is joined work with various partners from the LUNA project. Acknowledgement is provided to the respective groups where appropriate.

This remainder of this chapter is structured as follows: Methods and models are reviewed in Section 4.1, which includes a short discussion about the alignment problem. Section 4.2 describes methods for attribute value extraction, namely rule-based and statistical. Experimental results for the single systems are presented in Section 4.3 for the annotation with attribute names and in Section 4.4 for the additional extraction of attribute values. The possibility of reducing interpretation errors by combining some of the proposed methods is discussed in Section 4.5. A conclusion is given in Section 4.6.

4.1 Description of Modeling Approaches

In this section, various approaches to the task of concept tagging are presented. They include classical, well-known generative methods based on FSTs or SMT as well as discriminative methods like MEMMs and SVMs as well as techniques recently applied to natural language processing such as CRFs (discriminative) or DBNs (generative). For the closely related log-linear models MEMM and CRF, a detailed presentation is given in Section 1.4. Although discriminative models can (at least in principle) be converted to generative ones, as presented

in [Heigold & Lehnen+ 08, Heigold & Ney+ 11], they still have very different characteristics. In [Rubinstein & Hastie 97] it is concluded that generative models are cheap, but they need to fit well to the true distribution whereas discriminative training is more expensive but also more robust and special knowledge about the true distribution is not needed. Thus, it might be worth to evaluate the presented methods also based on the kind of model.
For consistency, the following naming scheme will be used throughout this chapter:

- **concept**: a set of attributes which is assigned to a sequence of words. This set contains up to two elements: the attribute name and the attribute value.
- **attribute name**: the tag representing the semantic meaning of the word sequence. The attribute name is required for each concept.
- **attribute value**: depending on the attribute name, there may be an associated normalized value which has to be extracted additionally from the word sequence.

4.1.1 Alignment

Except for the DBN approach, all presented methods rely on a one-to-one alignment between the input word sequence and the attribute name sequence, at least for training of the models. Therefore, the probability of the *concept* sequence $P(c_1^T|w_1^N)$ is projected to the probability of the *concept tag* sequence $P(t_1^N|w_1^N)$ by assigning "start" (s) and "continue" (c) markers to concepts. Here, the so-called begin insides outside scheme (BIO), proposed in [Ramshaw & Marcus 95], has been adopted. Using this approach results in a one-to-one alignment and the original attribute name sequence can be recovered. As a consequence, the segmentation given in the training data is also modelled:

$$P(t_1^N|w_1^N) = P(c_1^T, s_1^T|w_1^N) \tag{4.1}$$

It should be noted that the concept tags are just introduced for modeling reasons and do not appear in the final output of the systems. Fig 4.2 gives an example from the French MEDIA corpus [Bonneau-Maynard & Rosset+ 05] illustrating concepts versus concept tags.

In Figure 4.2 a, the attribute names as given with the corpus are shown. The respective concept tags, including the BIO notation realized with start/continue tags, are given in Figure 4.2 b. For example, the utterance part `chambre double` is mapped from the attribute name

$$\underbrace{\textbf{chambre-type}\{\texttt{chambre double}\}}_{c_3:w_4,w_5}$$

to the two concept tags

$$\underbrace{\textbf{chambre-type\_s}\{\texttt{chambre}\}}_{t_4:w_4}\ \underbrace{\textbf{chambre-type\_c}\{\texttt{double}\}}_{t_5:w_5}$$

using the start/continue scheme. A disadvantage of applying this mapping scheme is that the number of output symbols is roughly doubled. More examples from this corpus including the attribute value notation are given in Section A.1.0.1.

null{je veux} **nombre-chambre**{une} **chambre-type**{chambre double}

(a) Attribute name tags (concepts) for an example sentence.

null_s{je} **null_c**{veux} **nombre-chambre_s** {une}
chambre-type_s{chambre} **chambre-type_c**{double}

(b) Conversion to concept tags for the example sentence realizing a one-to-one alignment.

Figure 4.2 Example sentence from the French MEDIA corpus illustrating the mapping from concepts (attribute names) to concept tags. A translation of the French sentence would be "I would like a double room".

4.1.2 Stochastic Finite State Transducers - FST

The FST approach is a stochastic generative approach which computes the joint probability between the word sequence and the concept tag sequence. Since it is based on the paradigm generally used for ASR, this approach is well suited to process speech. An integrated decoding of speech and concept tags is also possible. Here, the task is to find the best concept tag sequence $\hat{t}$ by maximizing $p(t_1^N|x_1^T)$, where x_1^T denotes the acoustic signal. Finding the best concept tag sequence $\hat{t}_1^N$ given the acoustic observations x_1^T is formulated as:

$$\hat{t}_1^N = \underset{t_1^N}{\operatorname{argmax}} \left\{ \sum_{w_1^N} p(x_1^T|w_1^N,t_1^N)\; p(w_1^N,t_1^N) \right\} \tag{4.2}$$

$$\approx \underset{t_1^N,w_1^N}{\operatorname{argmax}} \left\{ p(x_1^T|w_1^N)\; p(w_1^N,t_1^N) \right\} \tag{4.3}$$

with

$$p(w_1^N,t_1^N) = \prod_{n=1}^{N} p(w_n,t_n|w_{n-1},t_{n-1},w_{n-2},t_{n-2}) \tag{4.4}$$

$$\tag{4.5}$$

According to Equation 4.3, two models are used for decoding. On the one hand, the acoustic model $p(x_1^T|w_1^N)$ which is given by the ASR system and on the other hand the joint probability of a word sequence and a concept tag sequence $p(w_1^N,t_1^N)$. This language model probability is modeled as a joint tri-gram model, which in some way can be compared to the graphoneme n-gram models used in G2P (cf. e.g. Section 6.1.4). The corresponding equation is given in Equation 4.4.

Typically, this decoding process is done sequentially: first, an ASR system is applied to generate the first-best hypothesis for the word sequence w_1^N. In a second step, the maximization

over the tag sequence t_1^N for $p(w_1^N, t_1^N)$ is carried out. Using this FST approach, it is possible to perform an "integrated" decoding, where not the first-best hypothesis is used for the word sequence w_1^N but a word-graph representation of the ASR search space. Finite state transducers using the AT&T FSM/GRM Library [Mohri & Pereira$^+$ 02] have been used. The final transducer is a composition of five transducers,

$$\lambda_{SLU} = \lambda_G \circ \lambda_{gen} \circ \lambda_{w2c} \circ \lambda_{SLM} [\circ \lambda_v] \tag{4.6}$$

These five finite state transducers are defined as follows:

λ_G represents $p(x_1^T | w_1^N)$ from Equation 4.3, i.e. a word graph or word sequence, encoded as a FST generated by an ASR module. For all reported experiments, only the single-best hypothesis is used.

λ_{gen} converts words to classes (e.g. cities, months, ...). The class representation models *a priori* knowledge of the task and allows for a better generalization on the training data.

λ_{w2c} translates words/phrases/classes to concept tags. This mapping is usually induced from the training data, but may also be performed using handwritten grammars (e.g. for dates or prices depending on the task). Additionally to the concept tags which have a semantic meaning for the task at hand, a filler concept tag is used to handle all utterances which do not convey a meaning.

λ_{SLM} denotes the stochastic trigram language model representing $p(w_1^N, t_1^N)$ from Equation 4.4 with classes. Note that to include the classes, the words have to be mapped to classes first.

λ_v converts the words tagged by an attribute name to a normalized attribute value. This is done using a rule-based approach (encoded as an FST) similar to the one described in Section 4.2.1. This step is optional.

The best sequence of concept tags is calculated as the best path of joint units of words and concept tags in the transducer λ_{SLU}. To obtain the concept tag sequence, a series of FST operations is performed:

$$\hat{t}_1^N = \text{nBest}(\pi(\text{project\_output}(\lambda_{SLU}))) \tag{4.7}$$

The output of λ_{SLU} is projected onto the concept tags (project_output) and unwanted symbols like <s>, </s> or the empty symbol ε are removed (denoted by π). For this work, only the single-best hypothesis has been used, but n-best hypotheses can naturally also be obtained by the nBest operation.

The FST approach, the training of the FSTs, the provision of hypotheses as well as the rules to map words to classes and to extract attribute values for the MEDIA corpus have been thankfully provided by Christian Raymond from University of Avignon (now with University Rennes 2 - Institut de Recherche en Informatique et Systèmes Aléatoires (IRISA)) and has been previously presented in e.g. [Raymond & Béchet$^+$ 06, Raymond & Riccardi 07].

4.1.3 Dynamic Bayesian Networks - DBN

DBNs have been often used in the literature to model various complex stochastic systems. In the last years, they have been used for many sequential data modeling tasks such as speech recognition [Zweig 98], part-of-speech tagging [Peshkin & Pfefer+ 03], dialog-act tagging [Ji & Bilmes 06], and in the domain of bioinformatics, e.g. for desoxyribonucleic acid (DNA) sequence analysis [Perrin & Ralaivola+ 03]. For this kind of graphical model and possible applications, an overview is given in [Murphy 02].

The DBN approach is also a generative directed graphical model and thus comparable with the FST approach and the involved probabilities. The main difference is the way in which both methods compute the probabilities and how training and decoding is performed. Additionally, DBNs do not rely on a one-to-one alignment of the training data. The following decision rule is used:

$$\begin{aligned} \hat{c}_1^T, \hat{v}_1^T &= \underset{c_1^T, v_1^T}{\operatorname{argmax}} \left\{ p(c_1^T, v_1^T | w_1^N) \right\} \\ &= \underset{c_1^T, v_1^T}{\operatorname{argmax}} \left\{ p(w_1^N | c_1^T, v_1^T)\ p(v_1^T | c_1^T)\ p(c_1^T) \right\} \end{aligned} \tag{4.8}$$

Both, the (best) attribute name sequence $\hat{c}_1^T$ as well as the (best) attribute value sequence $\hat{v}_1^T$ are decoded in parallel. To derive the attribute name sequence only, we have to sum up over all possible attribute value sequences (marginalization of Equation 4.8):

$$\hat{c}_1^T = \underset{c_1^T}{\operatorname{argmax}} \left\{ \sum_{v_1^T} p(w_1^N | c_1^T, v_1^T)\ p(v_1^T | c_1^T)\ p(c_1^T) \right\} \tag{4.9}$$

The attribute value extraction is performed under the same scheme and described in Section 4.2.2. Note that decoding is performed at the segmental level, i.e. there is an inner mechanism dealing with transitions between attribute name sequences. As presented in Equation 4.9, three probabilities or language models have to be estimated. Since all variables are observed in training, the probabilities can be derived directly from counts without the need of using e.g. the EM algorithm. The raw count estimates are improved using factored language models (FLMs) along with generalized parallel backoff (GPB), which are both presented in [Bilmes & Kirchhoff 03]. FLMs are an extension to classical n-gram LMs allowing to include more general features than just previous word occurrences. Unlike classical LM features, FLM features may appear at any time up to the time of prediction. Each word is associated with a vector of factors which may also include e.g. part-of-speech tags or attribute name/values. GPB is the extension of the classical back-off procedures to this new kind of feature vectors, which do not necessarily need to have a strict temporal order.

The three LMs used in Equation 4.9 are now realized as FLMs with some modelling assumptions. The probability for attribute name sequences is conditioned on the previous h words and

factorized:

$$p(c_1^N) = \prod_{i=1}^{N} p(c_i|c_{i-h}^{i-1}) \tag{4.10}$$

The probability of attribute values conditioned on attribute names is factorized:

$$p(v_1^N|c_1^N) = \prod_{i=1}^{N} p(v_i|c_i) \tag{4.11}$$

The probability for word sequences conditioned on attribute names is factorized and GPB with order w_{i-h}^{i-1}, c_i is applied to model the assignment to the attribute name c_i:

$$p(w_1^T|c_1^N) = \prod_{i=1}^{T} p(w_i|w_{i-h}^{i-1}, c_i) \tag{4.12}$$

Finally, the probability for word sequences conditioned on attribute names and values is factorized ; GPB is applied to model the assignment of the words to the attribute names and values and works with order w_{i-h}^{i-1}, c_i, v_i:

$$p(w_1^T|v_1^N, c_1^N) = \prod p(w_i|w_{i-h}^{i-1}, v_i, c_i) \tag{4.13}$$

Here, h represents the model's history which is varying between either bigrams or trirams in the systems used. GPB uses the modified Kneser-Ney discounting technique [Chen & Goodman 98] in all conditions. As already noted, for the DBNs used for the experiments in this work, the attribute name and value decoding steps are decoupled. First, the attribute names are decoded and then kept fixed for the attribute value extraction step, which is presented in Section 4.2.2.

The DBN approach, the training of the DBNs and the hypotheses themselves have been thankfully provided by Fabrice Lefévre from University of Avignon. The work has been previously presented in [Lefèvre 06, Lefèvre 07]

4.1.4 Phrase-based Statistical Machine Translation - SMT

A standard phrase-based machine translation (PBT) approach which comprises several generative statistical models is used. The incorporated models include phrase-based models in source-to-target and target-to-source direction, IBM-1 like scores at phrase level, again in source-to-target and target-to-source direction, a target language model, and additional word and phrase penalties. These seven models resp. penalties are log-linearly combined [Mauser & Zens$^+$ 06]:

$$\hat{t}_1^N = \underset{t_1^N}{\operatorname{argmax}} \left\{ \sum_{m=1}^{M} \lambda_m \log\left(p_m(t_1^N, w_1^N)\right) \right\} \tag{4.14}$$

Here, $\log(p_m(\cdot))$ represents feature functions (which are the aforementioned statistical models) and λ_m the corresponding scaling factors. These factors are optimized using some numerical algorithm in order to maximize translation performance on a development corpus. In this case, optimization of the scaling factors is done with respect to the concept error rate (CER) score

using the downhill simplex algorithm. In contrast to general translation models, reordering of the target phrases composing the translation is not needed for NLU.

There is a certain relation between the SMT approach and the log-linear models like CRF or MEMM as presented in Section 1.4. The feature functions in the case of SMT are statistical models which return float values, i.e. the features are no more binary. Merely seven parameters for the combination of the models are tuned in contrast to the millions of parameters used within e.g. CRFs.

The software used to provide the SMT hypotheses is an PBT RWTH in-house realization as described in [Mauser & Zens$^+$ 06]. More details about the SMT approach in general as well as pointers to literature are given in Section 1.1.

4.1.5 Support Vector Machines - SVM

Whereas the three models introduced in the previous subsections are all generative models, i.e. modeling the joint probability $p(t_1^N, w_1^N)$ directly, SVMs realize a discriminative approach modeling the decision boundaries between classes directly. A general introduction to SVMs is given e.g. in [Cortes & Vapnik 95]. This approach can be extended to model conditional probabilities $p(t_1^N|w_1^N)$ [Platt 99].

Using this (local) classifier-based approach, the task of tagging a sequence of words with attribute names is seen as a sequence of classification problems, one for each attribute name tag within the sequence. Therefore, the training data is represented as vectors within a high-dimensional feature-space. SVMs now maximize the geometric margin between the various classes while minimizing the classification error as a consequence. Various correlated and non-local features can be utilized, but it is not possible to trade off decisions at different positions against each other as with generative models.

For the reported experiments, the open-source toolkit Yet Another Multipurpose CHunk Annotator (YamCha) has been applied [Kudo & Matsumoto 01, Kudo & Matsumoto 05]. This SVM toolkit has been especially designed for chunking text which includes tasks like POS tagging, named entity recognition, or base noun phrase (NP) chunking (recognizing the chunks of a sentence which are noun phrases). It has been applied in the Conference on Computational Natural Language Learning (CoNLL) 2000 shared task on chunking and base NP chunking [CoNLL-2000 00] and performed best. Heuristic combinations of forward- and backward-moving sequential SVM classifiers are used taking into account previous decisions as features.

Since SVMs are binary classifiers, they need to be extended to multi-class classifiers for the task at hand. This is easily possible by building pairwise classifiers. For a K-class classification problem, all pairs of classes are considered, resulting in a total of $\frac{K(K-1)}{2}$ classifiers. The result of all of these classifiers is either combined by weighted voting leading to the final decision (*o*ne-versus-one strategy) or by letting the classifier with the highest output function decide (*one-versus-all* strategy).

The SVM approach, the training of the SVMs and the hypotheses themselves have been thankfully provided by Christian Raymond from University of Avignon (now with University of Rennes 2 - IRISA). The work has been previously presented in [Raymond & Riccardi 07].

4.1.6 Maximum Entropy Markov Models - MEMM

Both, the log-linear respectively discriminative MEMM and CRF approaches have been introduced in Section 1.4. For reference, they are recalled here shortly.

The MEMM model is an exponential model with a normalization on a positional level:

$$p_{\lambda_1^M}(t_1^N|w_1^N) = \prod_{n=1}^{N} \frac{\exp\left(\sum_{m=1}^{M} \lambda_m \cdot h_m(t_{n-1}, t_n, w_1^N)\right)}{\sum_{\tilde{t}_n} \exp\left(\sum_{m=1}^{M} \lambda_m \cdot h_m(t_{n-1}, \tilde{t}_n, w_1^N)\right)} \tag{4.15}$$

Here, $h_m(\cdot)$ represents the feature functions depending on a word sequence w_1^N and the corresponding attribute name tag sequence t_1^N. Training such a model is defined as finding optimal feature weights λ_m. All possible attribute name tags are represented by $\tilde{t}_n$. As feature functions, lexical features, the bigram feature on attribute name side, prefix and suffix features as well as a capitalization feature is applied. Decoding is performed by maximizing over the attribute name tags:

$$\hat{t}_1^N = \underset{t_1^N}{\operatorname{argmax}} \left\{ p_{\lambda_1^M}(t_1^N|w_1^N) \right\} \tag{4.16}$$

$$= \underset{t_1^N}{\operatorname{argmax}} \left\{ \prod_{n=1}^{N} \frac{\exp\left(H(t_{n-1}, t_n, w_1^N)\right)}{\sum_{\tilde{t}_n} \exp\left(H(t_{n-1}, \tilde{t}_n, w_1^N)\right)} \right\} \tag{4.17}$$

Whereas first experiments have been performed using the in-house MEMM software as provided by and used in [Macherey 09], most of the presented results have been produced using an in-house extension of the RWTH ASR framework which does also realize CRFs.

4.1.7 Conditional Random Fields - CRF

The CRF model is very similar to the MEMM model introduced in the previous section and basically differs only in the normalization term, which is on string level:

$$p_{\lambda_1^M}(t_1^N|w_1^N) = \frac{\prod_{n=1}^{N} \exp\left(\sum_{m=1}^{M} \lambda_m \cdot h_m(t_{n-1}, t_n, w_1^N)\right)}{\sum_{\tilde{t}_1^N} \prod_{n=1}^{N} \exp\left(\sum_{m=1}^{M} \lambda_m \cdot h_m(\tilde{t}_{n-1}, \tilde{t}_n, w_1^N)\right)} \tag{4.18}$$

The feature functions are the same as used for the MEMM approach; the decision rule is also similar, but due to the normalization on string level, the maximization is simplified:

$$\hat{t}_1^N = \underset{t_1^N}{\operatorname{argmax}} \left\{ p_{\lambda_1^M}(t_1^N|w_1^N) \right\} \tag{4.19}$$

$$= \underset{t_1^N}{\operatorname{argmax}} \left\{ \prod_{n=1}^{N} \exp\left(H(t_{n-1}, t_n, w_1^N)\right) \right\} \tag{4.20}$$

$$= \underset{t_1^N}{\operatorname{argmax}} \left\{ \sum_{n=1}^{N} H(t_{n-1}, t_n, w_1^N) \right\} \tag{4.21}$$

As for MEMMs, the training criterion is convex, but due to the complex normalization, CRFs are computationally expensive.

While initial experiments have been performed using the CRF++ toolkit [Kudo 05], the reported experimental findings are based on an in-house CRF realization within the RWTH ASR framework.

4.2 Attribute Value Extraction

In general, two ways could be considered to handle attribute value extraction. On the one hand, it is possible do directly hypothesize attribute names and attribute values in a single computational process. On the other hand, a hierarchical approach might be meaningful, where in a first step, the attribute names are hypothesized. In a second step, a model which extract the attribute values is applied to the already hypothesized attribute names and the corresponding word sequence, which could thus use the additional knowledge about the concept names and the segmented word sequences.

Which of these two ways is the better choice does also depend on the application domain and the complexity of the task at hand. Except for simple application domains, attribute values are characterized by different model types. Examples include dates, which can be well represented by regular expressions, while city names or other named entities (even compounds) can be better represented as single lexical items. For this reason, a hierarchical approach to the task of attribute value extraction might be useful. Another complication is that in certain cases, the attribute values are normalized, especially when they can be expressed with synonyms.

For example, in the sentence of the MEDIA corpus `I'd like a room for no more than fifty euros`, the word sequence `no more than` is associated with the attribute name **comparative-payment-room**, which maps this sequence to the normalized value [less-than].

Within the MEDIA corpus, three different model types can be distinguished, namely numeric units, proper names or semantic classes. For each of these model types, a different value extraction mode has to be applied:

- value enumeration, e.g.: the attribute name “comparative” with possible values [around], [less-than], [maximum], [minimum], [more-than]
- regular expressions, e.g.: dates, prices
- open values (i.e. no restrictions), e.g.: client’s names

Note that all tags in the original MEDIA corpus are in French and here only translated for reference. More details about the MEDIA corpus are given in the corpus description in Section A.1.0.1. For the Polish and Italian corpora, the occurring types of attribute values are quite similar. For example in the Polish corpus, which deals with public transportation queries in Warsaw, there are a large number of open values, e.g. street or bus stop names are often parts of queries as well as town names or even points of interest, like important buildings or places.

Naturally, time information is also important for such a system and thus there are many attribute names dealing with dates, hours and minutes, whose attribute value extraction can be well covered with regular expressions. For the Italian task, an IT help-desk application, open values include user names and surnames or phone numbers, whereas there are enumerations like the various institutions which are supported by the help-desk. For both corpora, more details can be found in Sections A.1.0.2 and A.1.0.3 respectively.

To solve this normalization step, again two approaches are commonly used. On the one hand, deterministic rules are derived and applied (either manual or by regular expressions, depending on the model type) or, on the other hand, the normalization can be learned with a stochastic model by introducing an additional level. Within this work, several approaches have been considered based on rules and stochastic models and their combination. In the remainder of this section, the various approaches used are presented.

4.2.1 Approaches based on Deterministic Rules

A very common approach to attribute value normalization / hypothesization is the application of hand-crafted rules. The advantage of such an approach is a very good coverage to the task at hand as well as the possibility to extend the set of rules if new attribute names / values should be added to the data. The big disadvantage with such an approach is the very high cost w.r.t. human labor and time to create such a set of rules. Additionally, the annotator has to be an language expert and also understand the underlying task specific semantics. Since mostly some context information is also encoded within the rules, this approach is not very robust to syntactically/semantically error prone input which is usually produced by erroneous ASR output. This is another major drawback w.r.t. the actual application in an NLU system. Usually, script languages are used to encode the normalization step in form of regular expressions, which depend on the attribute name. They are applied to convert the word phrases supporting an attribute name to a normalized value. A starting point to obtain such rules is usually the (manually annotated) training data. Within this work, manually crafted rules are available for all of the three corpora.

Although this approach is not used in the presented work, it is also possible to encode such rules into FSTs. This might be interesting if word graphs from the speech input are available. FSTs can be composed with these to derive graphs of attribute name / value hypotheses which can be kept for further processing [Raymond & Béchet$^+$ 06, Servan & Raymond$^+$ 06]. In this work, either the manual transcription is used as input for the attribute value extraction step or the single-best ASR hypothesis.

Experimental results with rule-based and stochastic approaches as well as their combination are presented in Section 4.4.

4.2.2 Stochastic Approaches

In principal, it is possible to integrate an attribute value extraction step into all of the stochastic models for attribute name extraction presented in this chapter. Although the various models have a very general formulation, the integration of such an attribute value normalization is

Action[Request]{chciałam}**BUS**[151]{linię sto pięćdziesiąt jeden}...
Action[Request]{I would like}**BUS**[151]{line one hundred fifty one}...

Figure 4.3 Example sentence from the Polish SLU corpus illustrating complications w.r.t. stochastic attribute value extraction. The English translation is given for reference only and is not part of the corpus.

very tied to the model. In this work, only the CRF approach is used with an integrated value normalization. Since the DBN approach does naturally support such an integration, it is also described in the following subsection shortly.

Although the design of rules is expensive because of the human experts which are needed, the use of rules outperforms the results obtained with just stochastic approaches. Nevertheless, a combination of a stochastic approach and a rule-based system may still lead to improvements, as presented with the CRF model below.

4.2.2.1 DBN

As already presented in Equation 4.8, Section 4.1.3, the standard decoding process hypothesizes the combined sequence of attribute names and attribute values. Since the additional conditioning on the attribute values in $p(w_1^N|c_1^T, v_1^T)$ does increase the computational complexity greatly, it is not practical for real-world applications. Thus, either suboptimal decoding strategies like beam search may be applied, although the performance is not satisfactory (cf. [Lefèvre 06]), or a hierarchical approach can be used. In this approach, it is assumed that the normalized values do not influence the attribute name extraction and thus the segmentation of the input sentence significantly. As presented in Equation 4.9, marginalization of the attribute value sequence v_1^T is performed and the best attribute name sequence is calculated first. Subsequently, v_1^T is hypothesized given the single-best attribute name sequence $\hat{c}_1^N$:

$$\hat{v}_1^T = \underset{v_1^T}{\operatorname{argmax}}\, p(w_1^N|\hat{c}_1^T, v_1^T)p(v_1^T|\hat{c}_1^T)p(\hat{c}_1^T) \tag{4.22}$$

4.2.2.2 CRF

Knowing the location and the attribute name of content words given by the attribute name extraction, normalized values are hypothesized in a successive step for most of the attribute names as e.g. in the example from the Polish corpus concerning a *Request* about bus *151* given in Figure 4.3.

Here, from the phrase `I would like`, the normalized value [Request] has to be extracted, which is quite challenging for a stochastic system, because the corresponding phrase is variable. Similarly, the value [151] has to be extracted from the word sequence `line one hundred fifty one`. Here, the system has to learn, that only the numerical part of the phrase is rele-

vant for the attribute value extraction, if the value is a number. Note that the English translation is given for reference only and is not part of the corpus.

A one-to-one mapping between words and attribute values like in attribute name extraction is not used, instead exactly one value is hypothesized per attribute name. Therefore, a second CRF model is trained on the word/attribute name pairs on source side and the attribute values on target side. Thus, search is constrained to the set of seen attribute values for each attribute name and exactly one attribute value is extracted per attribute name and supporting word phrase. Additionally, mixing of attribute values is not allowed. Concerning the CRF feature set, the same features are applied which are used for the attribute name extraction, i.e. lexical features on the predecessor, the current, and the successor word, the bigram feature, capitalization, and prefix features. For attribute names with a huge number of values, it is possible to reduce the search space only to a "null" or empty value, leaving the attribute value extraction to a rule based approach in a possible post-processing step.

In the reported experiments, CRFs for attribute value extraction have only been included in the CRF approach and always in combination with rules. One reason supporting this combination is that the number of possible values varies highly between attribute names. For example, always in the Polish corpus, the attribute name **Reaction** can take either the value [Confirmation] or [Negation] and is triggered by only few content words. In contrast, the value of **STREET_NUMB** can at least theoretically be any number. It is likely that not all these numbers appear in the training corpus, which is the only information source available for training models in purely data driven approaches. The numbers can be easily covered more or less completely with regular expressions.

More details about the attribute value extraction process using CRFs are given in [Lehnen & Hahn$^+$ 09].

4.3 Experimental Results: Attribute Name Extraction

For all results presented in this and the following two sections, the same data has been used to train and evaluate the systems. The corpus statistics for the French MEDIA, the Polish Warsaw transportation and the Italian IT help-desk tasks are presented in detail and with examples in Section A.1. For each of the presented methods, only the respective training part of the data set has been used as the sole knowledge source. Depending on the method, the development set (denoted as *dev* in all tables) has been used to tune parameters, while the evaluation set (denoted as *eva*) is only used to assess the performance and generalization qualities of the tuned systems on unseen data. Due to the very different nature of the used corpora due to the underlying languages and tasks, various peculiarities have to be taken into account, which will be addressed where appropriate.

The evaluation itself, i.e. the scoring of the hypotheses, has been performed using the National Institute for Standards and Technology (NIST) scoring toolkit [NIST 95], which is generally accepted within the community and has been used in many Advanced Research Projects Agency (ARPA) and European projects. The main error criterion which has been used is the so-called concept error rate (CER). It is defined as the percentage obtained with the ratio of the

sum of deleted, inserted and confused concepts (not concept *tags*) $\{\hat{c}_1^{\hat{N}}\}_1^T$ hypothesized in the test set, and the total number of manually annotated concepts $\{c_1^N\}_1^T$ used as reference:

$$\mathrm{CER}(\{\hat{c}_1^{\hat{N}}\}_1^T, \{c_1^N\}_1^T) = \sum_{t=1}^{T} \frac{\mathfrak{L}(\{\hat{c}_1^{\hat{N}}\}_t, \{c_1^N\}_t)}{|\{c_1^N\}_t|} \tag{4.23}$$

Here, $t = 1, \ldots, T$ denotes the sentences within the corpus, while $n = 1, \ldots, N$ denotes the sequence of concepts within a sentence. $\mathfrak{L}$ denotes the Levenshtein distance. The sentence error rate (SER) is also used as a secondary performance measurement. It is defined as the percentage of sentences whose complete semantic annotation is equal to the one in the corresponding reference:

$$\mathrm{SER}(\hat{c}_1^T, c_1^T) = \frac{\sum_{t=1}^{T} \delta(\hat{c}_t, c_t)}{T} \tag{4.24}$$

If the two strings $\hat{c}_t$ and c_t are identical, the δ-function returns zero and one otherwise. Note that the indexes iterating over the individual concepts within sentences have been dropped for better readability. The *NULL* concept, representing out of domain groups, is removed from reference and hypothesis prior scoring for both measures, the CER and the SER. Note that theses measures are comparable to the phoneme error rate (PER) and word error rate (WER) as used to evaluate the G2P systems and presented in Section 6.2.1.

As a first step, the various systems are trained and optimized on the development set. This basically includes training of multiple systems with various parameter settings to find the one with the best performance on the dev set. Since the choice of feature functions is essential for the performance of log-linear models, the training process of the CRF system will now be shortly described as an example. The basic features have already been introduced in Section 1.4.1. Since it is not feasible to test all possible combinations of features and window sizes, we stick to the following selection process, which usually leads to good results: first, the regularization term is tuned with a basic feature set consisting of lexicon features in a window of $[-1 \ldots 1]$ around the current word and the bigram feature. Afterwards, enlarged windows for source-n-gram features are tested. With the optimal source-n-gram window w.r.t. CER on the dev set, the gain of word part features is determined in a similar manner: For pre- and suffix features, the length is successively enlarged and the best performing lengths for pre- and suffixes are determined. The capitalization feature is simply enabled in one experiment. If there is an improvement in performance, it will be kept for the final system. Then, the word part features are combined according to their gain. In a last step, the margin-posterior is used for the training of the final CRF system. This modified training criterion is presented and analyzed in detail in Chapter 5. An exemplary feature build-up for the French MEDIA corpus is presented in Table 4.1.

By only including very basic features, e.g. the source-n-gram feature at the current position (t_n, w_n), which is sometimes referred to as "membership feature", and the bigram feature (t_{n-1}, t_n), the CER on the eva set is with 14.6% already in a very good range, at least when compared with other optimized methods as presented in Section 4.3. Introducing the additional source-n-gram features (t_n, w_{n-1}) and (t_n, w_{n+1}) leads to a tripling of the number of active

Table 4.1 Feature build-up of the CRF system on the French MEDIA corpus including the number of active features.

	number of	CER [%]		
features	features	Train	dev	eva
(t_n, w_n)	419,900	82.6	91.6	88.9
+ (t_{n-1}, t_n)	456,190	9.7	15.3	14.6
+ $(t_n, w_{n-1}) + (t_n, w_{n+1})$	1,247,730	3.9	13.1	12.4
+ capitalization, prefixes	1,683,210	3.5	12.8	11.5
+ margin-posterior	1,683,210	10.0	12.3	10.6

Table 4.2 Optimized feature setups used with CRFs on the three concept tagging corpora. For the lexical features, the window size and location is given, while for the capitalization and bigram feature a checkmark is used to denote its use. For pre- and suffix features, the respective lengths are denoted.

	chosen features					
corpus	lexical	bigram	capitalization	prefix	suffix	# features
French	$w_{n-1}, \ldots, w_{n+1}$	✓	✓	1...4	-	1,683,210
Polish	$w_{n-1}, \ldots, w_{n+1}$	✓	✓	1...4	1...4	6,926,983
Italian	$w_{n-3}, \ldots, w_{n+1}$	✓	-	1...6	1...6	1,424,291

features, but also lowers the CER significantly by 15% relatively down to 12.4%. The morphological motivated word part features could reduce the CER by another 7% relatively down to 11.5%. By re-training this system with the margin-posterior training criterion, the CER drops by another 8% relatively down to our final result of 10.6% CER. Note that the margin-posterior leads to a better generalization as the training error rate rises while the CER on dev and eva decreases.

The optimization process, most prominently the choice of features and window sizes, depends on the task and the language. Table 4.2 shows the different setups for the final CRF systems. For the respective MEMM models, the pre- and suffix features are slightly different and derived by optimization on the development corpora.

For the French MEDIA corpus, a comparatively small source-n-gram window, the capitalization feature and prefix features for prefixes of lengths one to four lead to the best results. Suffixes did not help. Naturally, all three systems include the bigram feature. For the Polish database, which consists of a much higher number of concept tags to be distinguished and a larger vocabulary on source side (cf. Table A.1), the number of features for the final system is roughly four times the size as of the French system. Interestingly, the best feature setup is the same as for the French task with the exception that suffixes of lengths one to four are additionally included. For the Italian task, which is the smallest of the three tasks w.r.t. training data size and number of concepts, the optimal source-n-gram feature window is not centered around the current position but shifted to the left. The capitalization feature does not help, but the pre-

and suffix features are both helpful and thus included with lengths one to six each. The total number of features is comparable to the French system, though a little bit smaller, which seems to be reasonable due to the smaller number of concepts.

Single best results have been produced by all of the six presented approaches for the development (dev) and the evaluation (eva) sets with and without attribute value extraction. The results on manual annotations indicated as *text input* and on ASR hypotheses indicated as *speech input* for the attribute name extraction only task on the dev and eva corpora for the French, Polish and Italian tasks are given in Table 4.3. As contrastive results, the table also contains system combination results which will be discussed in detail in Section 4.5.

The single systems are ranked according to their performance in attribute name extraction on the evaluation set on text input.

The CRF model leads to the best tagging performance on the MEDIA task (text input) with a CER of 10.6%. There is a gap in CER of more than 25% relatively between the CRF approach and SVMs, which is the next best method on MEDIA. The tagging performance of the non-CRF systems varies between 13.4% and 15.5% CER. A first comparison of SVM, FST and CRF for SLU on French and English corpora has been published in [Raymond & Riccardi 07] and a detailed comparison of five of the six techniques described in this chapter in [Hahn & Lehnen$^+$ 08b]. Within the latter publication, all methods except DBNs have been tuned and applied to an earlier version of the MEDIA corpus (text and speech inputs). The best single system (also CRFs) performed slightly worse with 11.8% CER (compared to 10.6%). Using ASR input, the respective number is 24.6% for the CRF system (compared to 23.8%). In comparison to the results presented in [Hahn & Lehnen$^+$ 08b], improvements in CER have been achieved for all systems, e.g. due to the introduction of categorization as an additional feature for the FST system. The categorization is realized by the use of 18 generalization classes including numbers, weekdays, country names, hotel names, etc. For the CRF system, the modified margin-base training criterion leads to improvements. A detailed error analysis on concept level has shown that four concepts are tagged (slightly) better by competing systems: `object` (e.g. hotel) and `date` by the FST system, `connectProp` by the SVM system and `payment` by the MEMM model.

With respect to the Polish task, the overall trend is similar. CRFs (21.5% CER) outperform the other methods, but by a smaller margin of roughly 2% CER relative w.r.t. the second best performing system on text input, which are FSTs (21.9% CER). There is a bigger gap in CER of roughly 15% relative to the third best performing method. The error rates are roughly twice as high as for the MEDIA corpus, which can be attributed to some degree to the larger number of concepts to be distinguished and the larger vocabulary on source side. Also, the Polish task is more complex and there is less training material available. Additionally, the corpus consists of human-human dialogues (annotated by linguists) which are in general more natural and thus more complex to learn for statistical approaches than the Wizard-of-Oz dialogues used in the MEDIA corpus.

Concerning the Italian task, again the CRF approach seems to perform best with a CER of 20.0%, but with a small margin. The overall trend is again similar to the MEDIA and the Polish corpora, and there is again a bigger gap in CER between the second and third best performing system of roughly 18% relative.

Table 4.3 Results for attribute name extraction for the various tagging systems on the French, Polish and Italian tagging corpora. This includes single system and system combination results (CER [%]) on the manually (text input) and automatically (speech input) transcribed dev and eva corpora. The WER for speech input for French is 30.3% on dev and 31.4% on eva, for Polish 39.5% on dev and 38.9% on eva and for Italian 28.5% on dev and 27.0% on eva. The best system on eva (tuned on dev) is indicated in boldface.

		text input		speech input	
	model	dev	eva	dev	eva
French	CRF	12.3	10.6	24.0	23.8
	SVM	14.2	13.4	27.1	25.8
	MEMM	15.8	13.7	26.6	26.4
	FST	16.1	14.1	28.3	27.5
	DBN	17.0	15.5	29.5	29.1
	SMT	16.0	15.1	28.4	29.0
	weighted ROVER	11.6	**10.2**	23.4	**23.1**
	FST re-ranking	10.7	11.3	24.5	24.3
Polish	CRF	21.0	21.5	53.6	**51.7**
	FST	20.5	21.9	58.3	57.9
	MEMM	24.0	25.1	58.0	57.0
	DBN	27.5	26.6	58.9	57.7
	SVM	26.2	27.3	59.1	58.1
	SMT	27.2	27.7	60.3	59.0
	weighted ROVER	18.7	**18.9**	53.5	52.9
	FST re-ranking	17.4	19.5	57.4	56.5
Italian	CRF	20.6	20.0	30.0	28.4
	FST	22.1	20.1	35.6	33.3
	SMT	25.0	25.0	35.0	33.7
	SVM	24.6	25.3	36.3	34.0
	DBN	24.3	25.7	33.6	32.1
	MEMM	24.6	27.3	33.2	33.3
	weighted ROVER	19.5	19.8	29.3	**27.5**
	FST re-ranking	19.3	**18.3**	31.3	29.2

In any deployed dialogue system, a speech recognition system is used to provide the input word sequence for the concept tagging module. Since ASR is always error prone, it is necessary to analyze the effect of ASR errors on the tagging performance. Therefore, we use an automatic transcription of the development and the evaluation corpora. For MEDIA, the ASR word error rate is 30.3% for dev and 31.4% for eva. The corresponding tagging results of all six systems on attribute name extraction are also given in Table 4.3. The performance is measured w.r.t. the same attribute name reference sequence as for text input. Concerning the different kinds of errors produced by the systems, there is roughly the same trend as for the manual transcriptions. The CRF approach performs best with a CER of 23.8%, whereas the second best approach (FST) leads to a CER of 25.8%. Across systems, the CER raises by a factor of approx. 1.7–2.3 for speech input compared to text input. An error analysis revealed that for two concepts the tagging performance degenerates heavily due to the introduced recognition errors:

- the concept `answer` is relatively short covering mainly the key words "oui" (yes), "non" (no) and "d'accord" (agreed) which have often been deleted by the ASRsystem;
- `payment` often corresponds to the currency word "euro" which is also often deleted or confused by non-content words;
- there are also concepts for which the tagging performance is comparatively stable, e.g. `object` which is often found next to a co-reference tag `coRef`.

For the Polish task, the results on ASR input are also given in Table 4.3. Due to the pretty high WER of the ASR system (roughly 40%), even the best performing CRF system gets a CER of 51.7% on the evaluation set considering attribute names only. Despite the high error rate, these results on ASR input show that the CRF approach is quite robust, since the second best performing system scores 57.9%, which is a relative drop in performance of approx. 10%.

The results for the Italian task are given in the same Table 4.3. Again, the whole picture is similar to French and Polish. CRFs lead to the best result for ASR input (28.4% CER), followed by the other systems with a clear gap of several percents.

4.4 Experimental Results: Attribute Value Extraction

Except for the CRF system, the attribute value extraction is performed in the same way for all systems using a rule-based approach. For CRFs, the procedure has been the following: on the development set, the stochastic and rule-based attribute value extraction is performed in parallel on the reference text input. The errors of both processes are compared and, for each attribute name, the extraction method with less errors is chosen, e.g. for the MEDIA corpus, 16 out of 99 attribute names are covered by rules, namely date and time expressions. For Polish, 94 out of 195 attribute names are covered using rules. Here, the overall confusion is higher due to the high number of attribute names within the corpus. Mostly date/time expressions, bus numbers and locations/places are extracted using rules. For the much smaller Italian task, for only 10 out of 43 attribute names rules are used, which cover user data like names or surnames, problem types or cardinal numbers. In general, rule-based approaches work better for enumerable types

Table 4.4 Comparison of rule-based and statistical attribute value extraction and their combination for the CRF approach on all of the three tagging corpora covered in this chapter (CER[%]). Besides the text and speech input, also reference input is considered, which is the correct sequence of concepts.

		reference input		text input		speech input	
	extraction method	dev	eva	dev	eva	dev	eva
French	rule-based	4.3	4.8	15.2	13.5	29.0	28.2
	statistical	5.3	5.2	16.4	14.0	29.5	28.0
	combination	2.6	3.5	**14.5**	**12.6**	**28.6**	**27.3**
Polish	rule-based	6.8	7.2	26.4	26.3	59.7	57.3
	statistical	13.9	14.3	29.2	29.8	61.8	59.9
	combination	4.8	5.3	**24.5**	**24.7**	**59.1**	**56.7**
Italian	rule-based	3.2	2.9	22.2	22.4	33.1	32.1
	statistical	4.8	4.6	23.0	22.5	**32.4**	**31.1**
	combination	2.1	3.4	**21.7**	**21.8**	32.5	31.3

like numbers or for items which can be listed and put into a category like names or places. A comparison of rule-based and statistical attribute value extraction and their combination on all of the three tasks is given in Table 4.4 for the CRF approach.

Within the table, besides the text and speech input, also reference input is considered, which represents the correct respectively manual annotated sequence of attribute names. For all languages, the rule-based approach outperforms the stochastic approach, at least if reference or text input is considered. For speech input, the gap between rules and the statistical approach is pretty small, due to the fact that the rules fail to correctly process erroneous input. There are potentially two error sources here. On the one hand, the ASR output could already contain errors leading to a wrong classification. On the other hand, the attribute name extraction step could be erroneous. This is also an indication that it might be meaningful to use statistic approaches for attribute value extraction which are more robust to erroneous input. For Italian, the statistical approach appears to perform slightly better on speech input than the one using rules, even if the advantage is small. For almost all input conditions and tasks, the combination of both approaches gives a significant gain in performance. Considering text input, the performance on MEDIA could be improved by roughly 6% relative due to combining rules and the statistic approach. The same is also true for the Polish task, which might seem a bit surprising, since the performance of the statistical approach is 50% relative worse on reference input. But since the quality of the stochastic approach varies highly with the attribute name, a combination still gives a considerable gain in performance. For the Italian task, the combination leads to a smaller gain of roughly 2% relative on text input and an insignificant loss in performance on speech input. Thus, the combination of stochastic approach and rules has been used for the CRF approach for all tasks/languages.

Results for attribute name and attribute value extraction for all of the six single systems on text and speech input are compared in Table 4.5.

Table 4.5 Results for attribute name and attribute value extraction for the various tagging systems on the French, Polish and Italian tagging corpora. Single system and system combination results (CER [%]) on the manually (text input) and automatically (speech input) transcribed dev and eva corpora. The WER for speech input for French is 30.3% on dev and 31.4% on eva, for Polish 39.5% on dev and 38.9% on eva and for Italian 28.5% on dev and 27.0% on eva. Numbers in brackets refer to a combination of statistical and rule-based attribute value extraction used only for the CRF approach. All other figures use the same rule-based approach.

		text input		speech input	
	model	dev	eva	dev	eva
French	CRF	15.2 (14.5)	13.5 (12.6)	29.0 (28.6)	28.2 (27.3)
	SVM	17.2	15.9	31.5	29.7
	MEMM	18.2	16.3	31.4	30.7
	FST	18.3	16.6	32.5	31.3
	DBN	19.3	17.4	34.6	32.8
	SMT	18.8	17.8	33.3	33.5
	weighted ROVER	13.8 (13.6)	12.0 (**12.0**)	27.8 (27.5)	27.0 (**26.0**)
	FST re-ranking	13.6	13.3	29.1	27.8
Polish	CRF	26.4 (24.5)	26.3 (24.7)	59.7 (59.1)	57.3 (**56.7**)
	FST	26.1	27.1	65.3	64.0
	MEMM	29.1	30.0	63.1	61.7
	SVM	30.3	31.2	63.3	61.5
	DBN	33.2	31.4	64.8	63.1
	SMT	33.6	33.6	66.2	64.4
	weighted ROVER	23.7 (23.2)	24.4 (**23.7**)	60.4 (58.6)	58.6 (57.2)
	FST re-ranking	22.6	24.1	62.5	61.3
Italian	CRF	22.2 (21.7)	22.4 (21.8)	33.1 (32.5)	32.1 (**31.3**)
	FST	24.2	23.1	39.4	37.2
	SVM	25.8	27.1	39.7	36.7
	DBN	26.2	28.9	37.2	36.3
	SMT	27.4	27.9	38.8	37.5
	MEMM	26.3	29.3	36.9	37.0
	weighted ROVER	20.8 (20.3)	21.4 (21.6)	32.2 (32.3)	31.3 (31.6)
	FST re-ranking	21.2	**20.9**	34.8	32.6

The systems are ranked according to best performance on the text input data on the evaluation sets. Best systems are indicated by boldface numbers. The numbers in brackets refer to results obtained with a combination of rule-based and statistical attribute value extraction. All other figures are obtained using only rule-based attribute value extraction. Thus, the figures not in brackets use the same knowledge sources and are all comparable to each other.

Considering the French task, CRFs outperform all other methods on text and speech input, even without considering the combined rule-based/stochastic attribute value extraction. On text input, the final result using rules only leads to a CER of 13.5%, which is roughly 15% relative better than the second best performing system (SVMs). On speech input, the difference between these two systems is approximately 5% relative. If the combined rule-based/stochastic attribute value extraction is considered additionally, the systems diverge further, resulting in a CER of 12.6% on text input and 27.3% on speech input, which is roughly 20% and 8% respectively better than the SVM system on text and speech input. Overall, the CRF model leads by far to the best tagging performance on the MEDIA evaluation corpus with 10.6% CER considering only attribute names and text input. If attribute values are additionally extracted (via a combination of rule-based and stochastic approaches; details are given below), a CER of 12.6% is achieved. Compared to the best result submitted to the MEDIA evaluation campaign in 2005 (19.6% CER, attribute name/value extraction, relaxed-simplified condition, cf. [Bonneau-Maynard & Ayache$^+$ 06]), this is a relative reduction of roughly 35%.

For the Polish task, the overall trend is similar as for the MEDIA task: the CRF model outperforms all other models with a CER of 24.7% for attribute/value extraction on text input. The second best performing system, FST, has a relative loss in performance of roughly 10% w.r.t. CRFs. It seems to tend to over-fitting, since it is much better on the dev sets than on the eva sets. Concerning speech input, the CRF system's result could also not be improved by system combination, which will be presented in detail in the following Section 4.5. It is also interesting to see that the ranking of systems does differ for text and speech input on the Polish task. While SVMs are the fourth best system on text input, it is actually the second best considering speech input.

On the Italian task, the picture is again similar to Polish and French. While the performance between CRFs and the second best approach, FSTs is roughly 7% relative, on speech input it is even roughly 16%. Again, the ranking of the systems differ between text and speech input. The second best system on speech input, DBN, is roughly 14% worse than the CRF system. As already for the Polish speech input, the CRF system for Italian could not be improved using system combination techniques.

Another interesting point is the ranking of the systems across languages. CRFs seem to be the method of choice, since it always outperforms the five other methods. SMT seems to be the weakest modeling approach. Altogether, the gap between the various models is pretty big: the drop in performance between the best and the weakest model on text input is roughly 38% for French, 36% for Polish and 28% for Italian. On speech input, the corresponding figures are 20% for French and Italian, and 14% for Polish (note that the error rates for Polish speech input are pretty high in general). While the MEMM system performed considerably well on French and Polish, it is the worst performing system on Italian.

All the presented results show that there is a need for further error reduction. Even if it is

Table 4.6 Attribute name and value CER for the six described systems on the MEDIA evaluation corpus (text input). The CER is also presented broken down in substitution, insertion and deletion errors.

	attribute name and value error rates [%]				
model	substitution	deletion	insertion	CER	SER
CRF	5.1	4.8	2.8	12.6	21.0
SVM	5.9	6.8	3.2	15.9	24.7
MEMM	6.5	7.0	2.8	16.3	25.4
FST	6.6	4.8	5.1	16.6	25.8
SMT	6.5	6.1	5.3	17.8	26.6
DBN	5.7	6.1	5.6	17.4	26.9

difficult to make an assessment without building a real system, it is very likely that any dialogue manager will have difficulties in deciding erroneous inputs (especially if the error rates are as high as roughly 60% CER as for Polish). While the best available sequence classifiers have been tested individually, system combination is now conceivable to take the best advantage of them all.

4.5 System Combination Results

In this section, two approaches to combine systems for dealing with multiple hypotheses are described and evaluated. First, the well-known recognizer output voting error reduction (ROVER) is evaluated on the MEDIA corpora. Afterwards, a re-ranking approach combining discriminative and generative methods is presented. Although usually the application of system combination techniques is straight-forward as long as the single systems are already available, another reason behind using system combination for concept tagging is given in Table 4.6

With a closer look at the different kinds of errors made by the systems (cf. Table 4.6), we observe an imbalance between the different kinds of errors across the various systems. For example, the MEMM system has a relatively low amount of insertion errors and a relatively large number of deletions. The FST system on the other hand has a comparatively low amount of deletions, while the number of substitution errors for DBNs is comparatively low. This is an indication that system combination may help to reduce the overall error rate.

4.5.1 ROVER

Motivated by the differences in tagging performance on some individual concepts for the six systems, we performed light-weighted system combination experiments using (weighted) ROVER, which is known to work well for speech recognition [Fiscus 97]. Since we currently only consider the single best output of each system, ROVER performs majority voting after alignment based on the Levenshtein edit distance of the sequences of concept hypotheses generated by all of the systems. The reference for the alignment is the most likely sequence according to the

best performing system, which are CRFs. Additionally, the system weights for ROVER are optimized on the dev set using Powell's method (*multistart*, i.e with multi-start initializations) [Powell 77]. Here, we start with ten runs in parallel with randomly chosen system weights. For the two best results, Powell's method is applied until convergence. Finally, the overall best system weights are chosen. This procedure is repeated for each development set, i.e. the weights are tuned for text and speech input as well as for attribute name extraction and for attribute name and value extraction. The system weights are then kept fixed for the respective evaluation set. The results are presented in the lower part of Tables 4.3 and 4.5 for attribute name and attribute name and value extraction for text and speech input for all three tasks.

Using all six systems on the MEDIA corpus, there is a relative gain of approx. 5% for text and speech input on the eva corpora (considering name and value pairs). We also tried to estimate system weights using the downhill simplex algorithm but there is no significant difference compared to Powell's method.

It should be noted that ROVER is rather robust as it improves the single-best system in all input conditions and improvements on the dev corpora always lead to improvements on the eva corpora.

For Polish, ROVER gives comparatively good results for text input. The relative improvements over the CRF system are roughly 12% for attribute names only and 4% for attribute name and value extraction on the eva corpora. Again, also the results on the dev corpora are better than the single-best system. On speech input, the results on the dev corpora are slightly better than single-best, but this does not carry over to the eva sets, presumably due to the overall high error rates. Additionally, the gap between the best and the second best system is also pretty big. In fact, also re-ranking, another system combination approach described in Section 4.5.2, does not lead to a gain over the CRF approach.

ROVER applied to the Italian task only generates statistically insignificant improvements (approx. 1% of relative improvement) on text input. On speech input, the picture is similar to Polish: the CER of the second best system is roughly 20% relatively worse than the CRF system. However, if only attribute name extraction is considered, ROVER leads to a small improvement of approx. 3% relative over the single-best system. If additionally attribute value extraction is performed, the ROVER result is comparable to the CRF result.

ROVER seems to be a good choice for robust system combination, since it is very easy and cheap to compute once the single-system outputs are available and leads to improvements in most cases. For the tasks, where the results are worse than single-best (Polish and Italian ASR input), however the loss in performance is not statistically significant.

To analyze how much gain would be theoretically possible using system combination techniques, we computed the oracle error rates for text input (cf. Table 4.7) for all corpora.

Concerning MEDIA, the oracle CER for the name and value condition is roughly half of the system combination result. This indicates that considering all system outputs provides a very high recall that can be exploited by a dialogue manager with potential improvements over the results obtained by just using system weights. For Polish and Italian, the figures are similar. For speech input, the oracle error rates only drop by 20–30% w.r.t. the single best system. This indicates that all systems have problems with erroneous input and to merely apply system combination techniques is not enough to improve performance.

Table 4.7 Additive oracle error rates (CER [%]) on the manually transcribed (text input) corpora for the six systems on the French, Polish and Italian corpora ordered by decreasing performance.

		attribute name		attribute name and value	
	model	dev	eva	dev	eva
French	CRF	12.3	10.6	14.5	12.6
	+SVM	9.4	7.7	11.1	9.2
	+MEMM	8.6	6.9	10.2	8.3
	+FST	6.8	5.5	10.1	8.3
	+SMT	6.1	4.9	9.1	7.4
	+DBN	5.0	4.3	7.2	6.4
Polish	CRF	21.0	21.5	24.5	24.7
	+FST	13.2	14.2	17.4	18.1
	+MEMM	11.8	12.8	16.0	16.6
	+SVM	10.6	11.6	14.7	15.4
	+DBN	9.4	10.3	13.7	14.2
	+SMT	8.7	9.5	13.0	13.5
Italian	CRF	20.6	20.0	21.7	21.8
	+FST	14.7	12.8	16.2	14.7
	+SVM	12.5	11.4	14.2	13.4
	+DBN	10.9	10.1	12.4	12.2
	+MEMM	10.1	9.5	11.7	11.6
	+SMT	10.1	9.1	11.6	11.2

4.5.2 Combination of Discriminative and Generative Algorithms (Re-Ranking)

The re-ranking approach described in this section builds upon the same observation as for the ROVER approach as described in the previous section, namely the difference in behavior of different systems. Here, the different characteristics of a discriminative model approach learning the conditional probability of a concept sequence given a word sequence and a generative model modeling the joint probability of a concept and a word sequence are combined. The reasoning here is that generative models tend to be more robust with respect to over-fitting on training data while discriminative approaches can model complex dependencies using various feature functions. Thus, it is likely that the combination of generative and discriminative models could bring improvements w.r.t. SLU by mixing characteristics of both models. The scheme applied here has been proposed in [Dinarelli & Moschitti$^+$ 09b] and does make use of two models which have already been introduced in Section 4.1. First, *n*-best lists are generated by the FST model which are ranked by the joint probability given by the stochastic language model (cf. Section 4.1.2). In a second step, this *n*-best list is re-ranked using a (discriminative)

SVM model. In this context, re-ranking is the process of providing an alternative ranking of the original n-best list. Particular kernels are used to achieve this task. The training and application of such discriminative re-ranking model is now shortly described.

The underlying model is a binary classifier which just returns the "most correct" hypothesis given two candidate hypotheses. These pairs of hypotheses are provided by the n-best list generated by the FST approach. To train such a classifier, the best hypothesis from the n-best list is selected by calculating the CER of all hypotheses w.r.t. the reference annotation and selecting the one with the lowest error rate. Let i be the position in the n-best list with the best hypothesis. All pairs of hypotheses $<\{\hat{t}_1^N\}_i, \{t_1^N\}_j>$ containing the best hypothesis as the first component are considered as positive examples for the binary classifier (i.e. returning 1). Negative examples are built by inverting the order of the hypotheses, i.e. $<\{t_1^N\}_j, \{\hat{t}_1^N\}_i>$. This is possible, since the kernel is symmetric. These examples will return 0. Given all theses pairs and the corresponding binary correctness value 0 or 1, the SVM can re-rank the n-best list based on correctness (see [Dinarelli & Moschitti$^+$ 09b] for more details).

For decoding, all possible pairs of hypotheses from the n-best list are build. Now, a way to compare *pairs* of hypotheses is needed. The kernel that has been used to evaluate pair similarity in the re-ranking model is the partial tree kernel (PTK) proposed in [Moschitti 06], applied to the semantic tree called FEATURES [Dinarelli & Moschitti$^+$ 09a]. Within this framework, various important SLU features are considered, e.g. concepts annotated by the FST model, concept segmentation and surface form of the input sentence together with some word features. For the MEDIA corpus, the features used in the tree include word categories. For the Italian corpus, similar generalization features were used, e.g. general categories like months, numbers, or dates as well as syntactic categories for articles, prepositions, adjectives and some adverbs, useful to generalize semantic head prefixes (e.g. `with my printer` becomes `PREP ADJ printer`). For the Polish corpus, only the surface form was represented in the tree structure without any additional features.

Now, pairs of trees are built starting from the n-best list provided by the FST model, as presented in [Dinarelli & Moschitti$^+$ 09a]. The following example is taken directly from this paper. Let us suppose that 10-best interpretations are generated by the FST model, whereas s_i is the interpretation at position i for $i \in [1,..,10]$ and s_j is the best interpretation among them. Positive instances for training are then built as pairs $e_k = <s_j, s_i>$ for $i \in [1,..,10]$ with $i \neq j$ whereas the negative instances will be $e_k = <s_i, s_j>$. Instances for classification are then built with all possible combinations of the n-best list $e_k = <s_m, s_n>$ for $m,n \in [1,..,10]$ with $m \neq n$. With an abuse of notation, let s_i denote also the tree built from the corresponding interpretation, the pairs of trees built from the n-best list are used to train the re-ranker using the following re-ranking kernel:

$$K_R(e_1,e_2) = PTK(s_1^1, s_2^1) + PTK(s_1^2, s_2^2) \tag{4.25}$$

$$- PTK(s_1^1, s_2^2) - PTK(s_1^2, s_2^1) \tag{4.26}$$

where s_k^i is the i-th tree of the k-*th* pair e_k and e_1 and e_2 are two pairs in the set of training instances. In decoding, this re-ranking kernel is computed on the n-best hypotheses. The n-best list is re-ranked according to this score and the new best hypothesis is retrieved.

This re-ranking model combining results from four kernels has been used before for comparing pairs of hypotheses in various tasks like semantic role labeling reranking [Moschitti & Pighin$^+$ 06], parse re-ranking [Collins & Duffy 02, Shen & Sarkar$^+$ 03] or machine translation [Shen & Sarkar$^+$ 04], and for SLU in [Dinarelli & Moschitti$^+$ 09b, Dinarelli & Moschitti$^+$ 09a].

The experimental results for the re-ranking model are presented in the lower part of Tables 4.3 and 4.5 together with the ROVER results described in the previous section. Here, a 10-best list has always been used. Compared to the single systems on attribute name extraction only (the first of the two tables referred to), the FST re-ranking outperforms the best system (CRF) on Polish and Italian text input. On the French MEDIA task, the system is close to the CRF system and better than the second best approach for both, text and speech input. In particular, re-ranking always outperforms FSTs and SVMs, the two models which are used to build the re-ranking model. Considering the Italian task, the re-ranking model leads to the best results and does even outperform ROVER on text input. But again on speech input, the results are worse than ROVER and CRFs. One explanation for the bad performance on speech input could be that the approach is penalized by the lack of robustness of the underlying FST model as well as its tendency to over-fitting. Especially on Polish, where no system is able to achieve CERs better than 50.0%, the improvement of the re-ranker over the FST baseline is only roughly 1.5% and 2.4% relative for dev and eva respectively, whereas for MEDIA the figures are 13.4% and 11.6% and for Italian 12.1% and 12.3% respectively. In general, discriminative models seem to have a better performance on erroneous speech input. When considering attribute name and value extraction, the trend is similar. For the MEDIA task, the CRF system is outperformed on text input and also on speech input, if only rule-based attribute value extraction is considered. But ROVER does outperform the re-ranking approach, most likely because it does make use of six different systems while the re-ranking model only combines two of those. On the Polish text input data, the CRF model is again outperformed as is ROVER if only rule-based attribute value extraction is considered for the included CRF model. On speech data, CRF and ROVER do both outperform the re-ranking approach. On the Italian task in the text input condition, the re-ranker gives the best results with an CER of 20.9%, which is an improvement of roughly 4% relative over the CRF model and 2% over ROVER. On speech input, the re-ranking approach is outperformed by CRF and ROVER.

The de-facto realization of the tree re-ranking model has been thankfully provided by Marco Dinarelli from University of Trento, Italy (now with LIMSI/CNRS, Paris, France). More details can be found in [Dinarelli 10, Dinarelli & Moschitti$^+$ 12].

4.6 Conclusions

In this chapter, we have presented six state-of-the-art models for concept tagging applied to three tasks of different complexity in different languages. Additionally, comparative results as well as results for system combination methods have been presented. The models have been applied in two conditions: manual transcriptions (text input) and automatic transcriptions provided by an ASR system (speech input). CRF has turned out to be the best performing

single-system on all tasks. Compared to previous publications, the CRF approach itself could be improved by using a margin extension to the training criterion which is described in the following chapter 5.

On the well-known French MEDIA corpus, a CER of 10.6% resp. 12.6%, if attribute value extraction is considered additionally to attribute name extraction, could be achieved on the evaluation set using manual transcriptions as input. This corresponds to a relative reduction of approx. 35% w.r.t. results in the literature. With automatic transcriptions, the comparable figures are 23.8% and 27.3%. Thus, when attribute values are additionally extracted, the CER raises by approx. 17–27% relatively. For Polish and Italian, there are no comparable figures available by other groups yet, since the corpora have been collected only recently. But a CER of 24.7% on eva for Polish text input, attribute name and value extraction, and 21.8% CER for the comparable figure in Italian seem to be a good start.

Applying ROVER system combination of all six models could further reduce the CER on most tasks. On French MEDIA, a 3–5% relative improvement could be achieved depending on the input condition. For Polish and Italian, ROVER could outperform the single-best system on text input whereas on speech input the performance is slightly worse. Overall, ROVER seems to be a quite robust approach to system combination.

The combination of generative and discriminative approaches has also been tested. An FST model has been combined with an SVM model within a tree re-ranking framework. Significant improvements over the underlying FST and SVM models could be achieved, especially on text input. Re-ranking is less robust on speech input which is mostly due to the missing robustness of the underlying FST system. In contrast to ROVER, this approach could be improved by re-ranking hypotheses from models which are considered more robust, like CRFs and by also taking into account larger n-best lists.

Additionally to the purely numerical findings, some general considerations can be made based on the results obtained with the proposed approaches and their comparisons:

- Attribute name (and value) extraction can be seen as a special form of translation from natural language into a meaning representation language. The best results have been obtained with CRFs, probably because the approach can handle the context of an entire dialog turn in an effective way using various feature functions modeling complex dependencies.

- Concerning attribute value extraction, handcrafted knowledge based on rules can be effectively combined with knowledge acquired with stochastic methods. Building rules requires a considerable effort and expert language/task knowledge. Nevertheless, general purpose semantic knowledge properly representing, for example, space and time entities and relations can be reused in many applications. This knowledge can also be used in e.g. feature functions of exponential models.

- Using multiple and different approaches results in hypotheses with different types of errors. By combining the hypotheses of various systems, the overall performance can be improved. The improvement might probably be increased by imposing additional constraints from a conversation context.

Word recognition errors introduced by ASR systems are particularly high with telephone applications involving real-world users. These errors effect semantically relevant words and phrases and are thus the reason for a large number of errors in attribute name and value extraction. Theses errors are due to background noise and multiple voices, failure in end-point detection, mispronunciation of words, difficulty in recognizing a large variety of proper names (OOV problematic) and other causes. In general, discriminative methods seem to lead to more robust results on speech input than generative approaches.

There are also errors in attribute name and value extraction on manual transcriptions of conversations as input, especially if real-world users are considered. Besides the typical errors which are induced by using statistical approaches, one reason might be that spoken language often does not follow the structure of written-style text, i.e. there are more or less arbitrary deviations, which are very difficult to learn from data, even if we consider manual transcriptions of speech as basis for the design and training of stochastic models.

Although there are still some problems, it is already possible to use automatic dialogue systems, at least for partial automation and for certain types of applications. Using confidence measures, it is possible to transfer sentences with a low confidence to a human operator. As confidence measure, e.g. posterior probabilities calculated on word lattices could be used. Especially for dialog systems, theses confidence measures should not only depend on the ASR hypotheses, but should include a certain coherence of the interpretation with the conversation history and with system prompts.

Chapter 5

Modified Training Criteria for CRF

In this chapter, some modifications to the standard training criterion for CRFs are introduced and compared on the various concept tagging corpora as used in the previous chapter and presented in Section A.1. This includes the introduction of a margin term as proposed in [Heigold & Schlüter+ 09] and a power approximation to the logarithm. The presented results have been previously published in [Hahn & Lehnen+ 09, Heigold & Dreuw+ 10]. Since the regularization constants might be influenced by changing the training criterion, optimized regularization parameters are also reported.

5.1 Standard Training Criterion

First, let us recall the standard training criterion for CRFs. Let $\{t_1^{N_r}, w_1^{N_r}\}_{r=1}^{R}$ be the labeled training data, realizing already a one-to-one alignment of concept *tags* t_1^N and words w_1^N using the BIO scheme. The standard training criterion for CRFs maximizes the entropy (MMI):

$$\mathscr{F}^{(MMI)}(\lambda_1^M) = \sum_{r=1}^{R} \log p_{\lambda_1^M}(t_1^{N_r}|w_1^{N_r}) - C \sum_{m=1}^{M} |\lambda_m|^p \tag{5.1}$$

with

$$p_{\lambda_1^M}(t_1^N|w_1^N) = \frac{\prod_{n=1}^{N} \exp\left(\sum_{m=1}^{M} \lambda_m \cdot h_m(t_{n-1}, t_n, w_1^N)\right)}{\sum_{\tilde{t}_1^N} \prod_{n=1}^{N} \exp\left(\sum_{m=1}^{M} \lambda_m \cdot h_m(\tilde{t}_{n-1}, \tilde{t}_n, w_1^N)\right)} \tag{5.2}$$

Typically, some regularization is added for a more stable convergence. We use the Lp-norm for $p \in \{0,1\}$ in Equation 5.1 with some normalization constant $C \geq 0$. For the comparison presented in this chapter, the feature weights of the training criteria are optimized using RProp [Riedmiller & Braun 93] as outlined in Section 7.3.2. For the default setting $p = 2$ (L2-norm), the optimization algorithm is expected to be stable w.r.t. the result, since the training criterion remains convex.

5.2 Modified Training Criteria

Next, different modifications to this standard training criterion are investigated. Both proposed modifications to the standard training criterion are instances of the unified training criterion

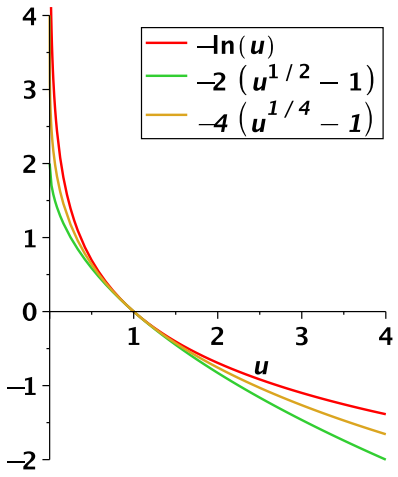


Figure 5.1 Examples for the power approximation to the logarithm. Besides the original logarithm, the curves of the approximation for $\xi = \frac{1}{2}$ and $\xi = \frac{1}{4}$ are shown.

(cf. [Macherey & Haferkamp+ 05]) and can thus be smoothly integrated into our transducer-based discriminative training framework as described in [Heigold & Schlüter+ 09].

5.2.1 Power Approximation to the Logarithm

For the standard training criterion in Equation 5.1, small class posterior probabilities are assigned a high loss. This is because the logarithm diverges for zero probabilities, $\log u \overset{u \to 0}{\to} \infty$. This means that the standard training criterion in Equation (5.1) is not robust against outliers, e.g. incorrect transcriptions. To avoid the divergence of the logarithm, the identity

$$\log u = \lim_{\xi \to 0} \frac{u^\xi - 1}{\xi} \tag{5.3}$$

is used to approximate the logarithm. In contrast to the logarithm, this approximation is bounded below for $\xi > 0$. This approximation is termed *power approximation* and resembles an error-based training criterion. The effect of this approximation is that bad outliers, which usually get weights assigned close to $log(0) = \infty$, are assigned much smaller weights far from infinity for accumulation. For this reason, this training criterion is expected to perform more robustly than the standard training criterion. Like all bounded/error-based training criteria for log-linear models (without proof), this training criterion has the disadvantage of not being convex. In our transducer-based framework supporting the unified training criterion [Heigold & Schlüter+ 09], the smoothing function $\log u$ for the standard training criterion is replaced with $\frac{u^\xi - 1}{\xi}$. In Figure 5.1, the behavior of the approximation is presented for some example values of ξ.

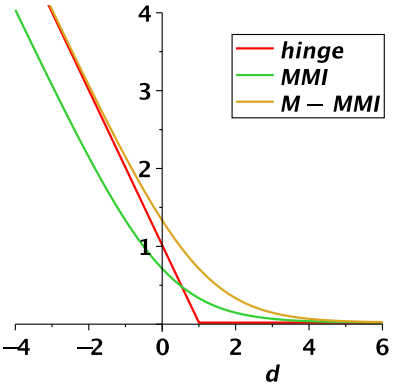


Figure 5.2 Illustration of the loss functions for SVMs (hinge loss), the original CRF training criterion (MMI) and the modified MMI (M-MMI), which includes a margin extension. For this example, a binary classifier has been used: $d := \sum_{i=1}^{I} \lambda_i (f_i(\tilde{t}_1^N, w_1^N) - f_i(t_1^N, w_1^N))$. Here, $\tilde{t}_1^N$ denotes the correct class (truth), while the competing class is denoted as t_1^N.

5.2.2 Margin-based Extension

A margin term can be incorporated into the standard training criterion as introduced in [Heigold & Schlüter+ 09]. A reason why this might be meaningful is that maximum-margin classifiers like SVMs and discriminative methods like CRFs do mainly differ in the loss function. For SVMs, the hinge loss is usually used (as an equivalent formulation for the modeling with support vectors and thus directly optimizing the margin without considering the logarithm or probabilities) modeling the joint probability $p(t_1^N, w_1^N)$ directly, SVMs realize a discriminative approach modeling the probability $p(t_1^N | w_1^N)$, which can be denoted as $\max(\rho - (t_1^N, \tilde{t}_1^N))$, while for CRFs the loss function is usually $log(p)$. The extended training criterion, referred to as margin-based MMI (M-MMI), can be interpreted as a smooth approximation to the hinge loss. The main difference between MMI and M-MMI is a shift in d, as is illustrated in Figure 5.2.

The M-MMI is realized by changing Equation 5.2 into a margin-posterior [Heigold & Schlüter+ 09]:

$$p_{\lambda_1^M,\rho}(t_1^N | w_1^N) = \frac{\prod_{n=1}^{N} \exp\left(\sum_{m=1}^{M} \lambda_m \cdot h_m(t_{n-1}, t_n, w_1^N) - \rho\, \mathscr{A}(t_1^N, \bar{t}_1^N)\right)}{\sum_{\tilde{t}_1^N} \prod_{n=1}^{N} \exp\left(\sum_{m=1}^{M} \lambda_m \cdot h_m(\tilde{t}_{n-1}, \tilde{t}_n, w_1^N) - \rho\, \mathscr{A}(\tilde{t}_1^N, \bar{t}_1^N)\right)} \tag{5.4}$$

with

$$\mathscr{A}(t_1^N, \tilde{t}_1^N) = \sum_{n=1}^{N} \delta(t_n, \tilde{t}_n) \tag{5.5}$$

Here, the margin score $\mathscr{A}$ is set to the word accuracy between the hypothesis t_1^N and the truth $\tilde{t}_1^N$, scaled with the factor $\rho \geq 0$ for smoothing. For $\gamma \to \infty$, the hinge loss is obtained.

The margin-based training criteria are obtained by replacing the posterior in Equation 5.1 by the margin-posterior in Equation 5.4. The such modified training criterion again fits into

Table 5.1 Concept Error Rates (CER) (attribute name extraction only) for various training criteria on the French and Polish SLU corpora.

	French		Polish	
training criterion	dev	eva	dev	eva
logarithm	12.8	11.5	21.8	22.6
power approximation	12.8	11.3	21.8	22.5
margin & logarithm	12.5	10.6	21.1	21.5
margin & power approximation	12.3	10.7	20.9	21.2

our transducer-based framework because the margin score can be incorporated by a composition [Heigold & Schlüter$^+$ 09]. The resulting training criterion is again convex, which is an advantage with respect to the feature weight optimization.

5.3 Experimental Results

We have experimentally tested three variants of the standard training criterion for CRFs on the three LUNA corpora which are described in Section A.1, considering attribute name extraction only. All setups were optimized from scratch. The experimental results are summarized in Table 5.1.

The experiments based on the power approximation in Equation 5.3 suggest that robustness is not an issue for these three corpora, probably because of the careful transcriptions of the data. On the French MEDIA and Polish corpora, there is no difference in CER on the dev corpora, and only small improvements on eva of 0.2% and 0.1% CER respectively. The incorporation of a margin term into the standard criterion, denoted as "margin & logarithm" in the table leads to consistent improvements, in particular on the evaluation corpora for both languages. For French, the CER on eva drops by roughly 8% relative. The Polish task benefits less from the margin term, by roughly 3% relative. This might be due to the increased confusability caused by the significantly larger vocabulary compared with the French task. For numerical reasons and similar to SVMs, the margin parameter ρ was set to unity and only the regularization constant was tuned. The optimal regularization constant for the margin-based training criteria tended to be smaller than for the corresponding training criterion without margin, for all tasks around 0.1, cf. Figure 5.3. Combining the power approximation with the margin extension ("margin & power approximation"), again does not help. For French, it even leads to worse error rates than with the original logarithm while for Polish, there is only a small improvement of 0.2% CER. An explanation for this observation might be that in contrast to the log-based criteria, the criteria based on the power approximation are non-convex and thus can get stuck in spurious local optima.

The margin extension together with the logarithm seems to be a robust choice and does improve the CER on all tasks. Thus, this training criterion has been chosen for the attribute name and value extraction task.

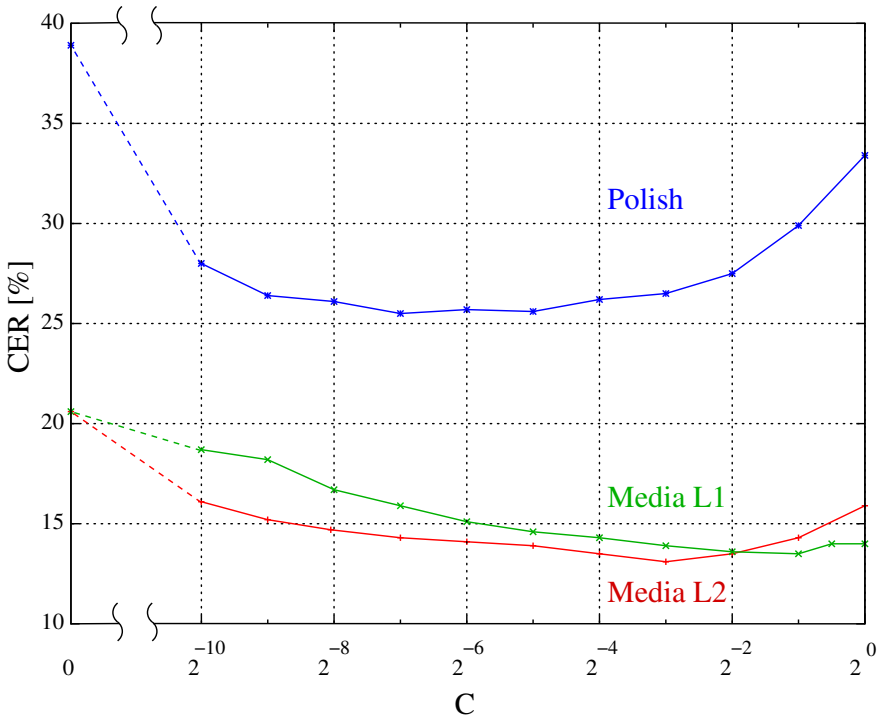


Figure 5.3 Tuning of the regularization parameter on the French Media and Polish dev corpora. For the French MEDIA corpus, *L1*-norm regularization is given in addition to *L2*-norm regularization.

5.3.1 Regularization

Two regularization variants, namely *L2*-norm and *L1*-norm, are widely used within CRFs. In Figure 5.3, the CER is given for various parameter settings.

On the French MEDIA corpus, both variants have been used and tuned individually resulting in lower error rates for the *L2*-norm regularization (CER of 13.1% on the dev set for the *L2*-norm versus a CER of 13.5% for the *L1*-norm). Based on the results on the French MEDIA corpus, only *L2*-norm regularization has been optimized for the Polish corpus. The regularization usually only has an effect on the CER when varied in an exponential manner. We have evaluated the range of the regularization parameter C from 2^{-11} to 2^{0}. Best error rates are obtained for $C = 2^{-3}$ and $C = 2^{-6}$ with a CER on the development set of 13.1% and 25.7% for French and Polish respectively (cf. Figure 5.3 for the complete curves). Note that both, *L1* and *L2* regularization can be used together to form the so-called elastic net (EN), which might be used for tasks where a filtering of feature functions has to be used as e.g. presented in Section 7.3 for grapheme-to-phoneme conversion.

As already discussed, our CRF modelling approach relies on a one-to-one mapping between word sequence and corresponding attribute name sequence. Using the BIO scheme, the attribute names are usually broken down with “start” and “continue” tags. Let `A` and `B` be two attribute names. In general, our CRFs implementation permits an attribute name tag sequence `A_start`

A B during search, which can not be seen in training, since it conflicts with the *start tag* rule, which would enforce the latter sequence to be A_start A B_start. This mismatch can be tackled by either interpreting a transition A → B as A → B_start or reducing the search space by all conflicting transitions like A → B. On both corpora, better results have been obtained for a range of regularization parameters by interpreting a transition A→B as A →B_start.

Chapter 6

Methods for Grapheme-to-Phoneme Conversion - A Comparison

In this chapter, we will take a closer look at various generative methods which are commonly used in real-life grapheme-to-phoneme conversion (G2P) applications [Hahn & Vozila+ 12]. Although most of the methods have already been published and/or are available as open source software, the reported experiments are done on large state-of-the-art tasks and the used software is from the actual publications. The work presented in this chapter has been performed at Nuance[1].

As already discussed in Chapter 1, Section 1.3.8, grapheme-to-phoneme conversion is usually used within every state-of-the-art ASR system to generalize beyond a fixed set of words. Although the performance is typically already quite good ($< 10\%$ phoneme error rate) and pronunciations of important words are checked by a linguist, further improvements are still desirable, especially for end user customization.

Besides an experimental comparison on text data for a range of languages (i.e. measuring the G2P accuracy only), our focus in this chapter is measuring the effect of improved G2P modeling on LVCSR performance for a challenging ASR task. Additionally, the effect of using n-Best pronunciation variants instead of single best is investigated briefly.

Over the years, many methods have been published to tackle the grapheme-to-phoneme conversion task. As already presented in Chapter 1, Section 1.3.8, this task is usually defined as follows: Given an orthographic form of a word (grapheme sequence $\mathbf{g}$), the corresponding most likely pronunciation (phoneme sequence $\boldsymbol{\varphi}$) is:

$$\boldsymbol{\varphi}(\mathbf{g}) = \underset{\boldsymbol{\varphi}' \in \Phi^*}{\operatorname{argmax}}\, p(\mathbf{g}, \boldsymbol{\varphi}') \tag{6.1}$$

Here, a grapheme $g \in \mathbf{g}$ is defined as a symbol used for writing language (e.g. a letter) and a phoneme $\varphi \in \boldsymbol{\varphi}$ as the smallest contrastive unit in the sound system of a language.

G2P is a task from the group of monotone string-to-string translation problems, which also includes POS, name transliteration [Deselaers & Hasan+ 09], and concept tagging (NLU) [Hahn & Dinarelli+ 11]. The application area is most prominently speech recognition as well as speech synthesis. Additionally, G2P is a tool which could be used for dictionary verification [Vozila & Adams+ 03] or to merge dictionaries with different phoneme sets (phone set mapping) [Chen 03].

[1] http://www.nuance.com/

Most of the published, statistical approaches to the G2P task can be decomposed into three sub-problems. As training material, usually a corpus is given containing corresponding pairs of orthographies and phoneme sequences. In a first step, an alignment is generated between graphemes and phonemes, since it is usually not provided within the training data, e.g.:

"phoenix"	=	ph	oe	n	i	x
[fiːnɪks]		[f]	[iː]	[n]	[ɪ]	[ks]

The resulting blocks of aligned tokens are typically referred to as joint-multigrams, grapheme-phonemes, graphonemes, or graphones for short in the literature and have been introduced in [Deligne & Yvon+ 95]. The length for the number of graphemes/phonemes per graphone may be restricted and empty tokens may be allowed on either side, depending on the alignment algorithm.

This alignment may be calculated prior to the training of the statistical model and kept fix or a re-alignment step may be (implicitely or explicitely) included within the training process which would be the next step. Here, mostly methods based on n-grams are used. The final step is the decoding, which determines how a given model is used to generate a phoneme sequence given a grapheme sequence.

One of the requirements of a G2P system for the presented comparison was that training and decoding can be done in reasonable time on large data sets. Thus, we did not use computational expensive methods like discriminative methods based on e.g. CRFs [Hahn & Lehnen+ 11] or online discriminative training as presented in [Jiampojamarn & Kondrak 09], or the recent extension of the phonetisaurus method with recurrent neural networks as presented in [Novak & Dixon+ 12].

The remainder of this chapter is structured as follows: in the next section, we will present the theoretical background to the five methods which we have compared. The following section will present experimental findings, both on text data only as well as integrated into a contemporary LVCSR system. The chapter concludes with a summary of our findings as well as an outlook.

6.1 Methods

In this section, we present the five methods used for the experimental comparison. The technical background is presented briefly with pointers to reference publications except for the first method which has been only documented in an in-house technical report. We applied the actual software used in the referenced publications, which in some cases is also available to the public (open source).

6.1.1 ngdt - Combined n-Gram and Decision Tree Model

Within this in-house method proposed in [Kneser 00], a classical graphone-based n-gram model is interpolated with a model based on decision trees. The alignment between graphemes and

phonemes is generated using a variant of the Baum-Welch EM algorithm. Instead of Baum-Welch, the Viterbi approximation has been tested, but there was not really any difference in the experimental results. Thus, Baum-Welch has been chosen since it is expected to be more robust and may even converge to the global optimum. The initialization resembles the first Baum-Welch iteration with uniform initial distributions. To obtain these distributions, all possible alignments not mixing deletions and insertions are averaged. Another common initialization would be to only take the word/pronunciation pairs where a 1-to-1 alignment suffices. Since there was no difference in the experimental results, the uniform distribution has been chosen.

Another constraint is that, for simplicity, only 1-to-*N*, e.g. grapheme-*unit* phoneme-*sequence* alignments are allowed. After convergence, the alignment and thus the resulting graphones are kept fixed for the actual model training.

To build the *n*-gram model, ML estimators are used on graphone sequences $\mathbf{q}$:

$$Pr(\mathbf{g},\boldsymbol{\varphi}) = Pr(\mathbf{q}) = \prod_{i=1}^{N} Pr(q_i|q_{i-1},\ldots,q_1) \tag{6.2}$$

$$= P_{ng(i)}(q_i|q_{i-n+1},\ldots,q_{i-1}) \tag{6.3}$$

To incorporate lower-order *n*-grams, their ML estimators are linearly interpolated using normalized interpolation scales α_i:

$$P_{ng}(\cdot) = \prod_{i=0}^{N} \alpha_i P_{ng(i)}(\cdot), \qquad \text{with} \sum_{i=0}^{N} \alpha_i = 1 \tag{6.4}$$

Within the (binary) decision tree, each leaf node C represents a set of samples, where a sample S is a 1-to-*N* pair from the aligned lexicon. A question about the context of the sample is associated with each non-leaf node. Two child nodes represent the positive and negative cases to this question. A sample can then be classified by traversing the tree and answering the questions at each node until the leaf node $C(S)$ is reached. For this node, we calculate

$$P_{dt}(\boldsymbol{\varphi}|C) = \frac{N(\boldsymbol{\varphi},C)}{N(C)}, \tag{6.5}$$

where $N(C)$ represents the total number of samples in C and $N(\boldsymbol{\varphi},C)$ the number of samples in C producing $\boldsymbol{\varphi}$. As training criterion, the entropy h is used:

$$-h := \sum_{S} logP_{dt}(\boldsymbol{\varphi}(S)|C(S)) \tag{6.6}$$

$$= \sum_{C}\sum_{\boldsymbol{\varphi}} N(\boldsymbol{\varphi},C)\ logN(\boldsymbol{\varphi},C) - \sum_{C} N(C)\ logN(C) \tag{6.7}$$

The idea now is to find a tree which describes the training data best. We start with a trivial tree consisting of one node and grow the tree by splitting leaf-nodes using a question from a given set of questions. At each split, the leaf node and question maximizing the entropy is selected.

Finally, the n-gram and decision tree model are log-linearly combined, whereas the interpolation parameter α is chosen empirically:

$$log(P_{ngdt}(\boldsymbol{\varphi}|\mathbf{g})) = \alpha\, log(P_{ng}(\boldsymbol{\varphi}|\mathbf{g})) + (1-\alpha)\, log(P_{dt}(\boldsymbol{\varphi}|\mathbf{g})) \tag{6.8}$$

An overview about decision tree models for grapheme-to-phoneme conversion is also given in [Yvon 96, Kienappel & Kneser 01, Bisani & Ney 08].

6.1.2 ibm - IBM joint ME n-gram model

In [Chen 03], a conditional as well as a joint ME n-gram model is used to tackle the G2P task. The latter one outperforms the conditional ME model and is very similar to the n-gram model presented in the previous subsection (cf. Equation 6.2):

$$Pr(\mathbf{g}, \boldsymbol{\varphi}) = \sum_{\substack{\mathbf{q}:\mathbf{g}(\mathbf{q})=\mathbf{g},\\ \boldsymbol{\varphi}(\mathbf{q})=\boldsymbol{\varphi}}} Pr(\mathbf{q}) \tag{6.9}$$

$$Pr(\mathbf{q} = q_1, \ldots, q_N) = \prod_{i=1}^{N} Pr(q_i|q_{i-1}, \ldots, q_1) \tag{6.10}$$

$$\tag{6.11}$$

The training schedule is somewhat different though. An ME n-gram model smoothed with a Gaussian prior is used to estimate $Pr(q_i|q_{i-1}, \ldots, q_1)$. First, a unigram model is trained on the graphones with the conventional Baum-Welch EM algorithm. As graphones, only 1-to-1 aligned grapheme/phoneme units including the empty token on both sides are allowed. For all following iterations, Viterbi EM is applied increasing n by one. Features are added to the model for all n-grams occurring in the Viterbi chunking of the training data. The training procedure is continued until convergence of the model, realigning the data in each iteration.

In the aforementioned paper, the author argues that the trivial graphone vocabulary is sufficient since n-grams allow for an intelligent modelling of context dependent phenomena, especially when well performing smoothing techniques are applied as described in [Chen & Rosenfeld 00]. Thus, high-performance models may be obtained, even for large orders of n.

6.1.3 dra - Dragon Joint n-gram Model

Within the method proposed in [Vozila & Adams$^+$ 03], the alignment is determined in a preprocessing step. N-to-1 graphones are selected via an HMM mechanism. Starting from uniform distributions, ML phoneme model distributions are estimated using the Baum-Welch algorithm. After model training, Viterbi is used to find the single-best alignment. A concatenative unit refinement step is possible by joining the m highest-ranking graphone pairs sorted by bigram frequency, although there are no improvements within the reported experimental results. For the actual model training, the joint probability $Pr(\mathbf{g}, \boldsymbol{\varphi})$ is calculated in the following way:

$$Pr(\mathbf{g},\boldsymbol{\varphi}) = \sum_{\mathbf{q}\in S(\mathbf{g},\boldsymbol{\varphi})} Pr(\mathbf{q})$$
$$= \max_{\mathbf{q}\in S(\mathbf{g},\boldsymbol{\varphi})} \prod_{i=1}^{N} p(q_i|q_{i-1},\ldots,q_1) \tag{6.12}$$

Here $S(\mathbf{g},\boldsymbol{\varphi})$ denotes the set of all co-segmentationss of $\mathbf{g}$ and $\boldsymbol{\varphi}$. The decoding is finally done using a best-first multi-stack algorithm, which is an approximation to the joint probability. For more details, the reader is referred to the original publication.

6.1.4 seq - RWTH Joint *n*-gram Model

Within the Sequitur G2P toolkit [Bisani & Ney 08], again a joint n-gram model is used. The joint distribution $p(\mathbf{g},\boldsymbol{\varphi})$ is reduced to a distribution over graphone sequences $p(\mathbf{q})$ which is modeled by an M-gram sequence model:

$$p(\mathbf{g},\boldsymbol{\varphi}) = p(q_1^K) = \prod_{i=1}^{K+1} p(q_i|q_{i-1},\ldots,q_{i-M+1}) \tag{6.13}$$

The graphonemic M-gram model $p(q_i|q_{i-1},\ldots,q_1)$ is estimated using ML EM training on an existing pronunciation dictionary. For this procedure, no prior letter to phoneme alignment is needed as the set of graphones Q is inferred automatically:

$$p(\mathbf{g},\boldsymbol{\varphi}) = \sum_{\mathbf{q}\in S(\mathbf{g},\boldsymbol{\varphi})} p(\mathbf{q}) \tag{6.14}$$

Here, $\mathbf{q} \in Q$ is a sequence of graphones and $S(\mathbf{g},\boldsymbol{\varphi})$ denotes the set of all possible co-segmentations of $\mathbf{g}$ and $\boldsymbol{\varphi}$. The author argues that since within ML training, a new graphone may not emerge once its probability is zero, a uniform initial distribution over all graphones with certain length constraints w.r.t. graphemes and phonemes is necessary. The inverse of the total number of possible graphones is used as initial distribution, with manually defined length constraints L:

$$p_0(q) = \left[\sum_{l=0}^{L}\sum_{r=0}^{L} |G|^l |\Phi|^r\right]^{-1} \tag{6.15}$$

Here, G and Φ denote the grapheme and phoneme sets respectively. Further, since ML estimates tend to over-fitting on the training data, trimming and smoothing techniques are applied. Whereas the applied evidence trimming has already been described in [Bisani & Ney 02], the smoothing of fractional counts is described in [Bisani & Ney 08]. For more details w.r.t. the modeling, the reader is referred to the original publication.

Altogether, the algorithm used has basically two parameters, i.e. the maximum number of letters/phonemes per graphone L and the span of the M-gram model. Concerning decoding, for the possibly non-unique segmentation into graphones, a maximum approximation is applied:

$$Pr(\mathbf{g}, \boldsymbol{\varphi}) \approx \max_{\mathbf{q} \in S(\mathbf{g}, \boldsymbol{\varphi})} p(\mathbf{q}) \tag{6.16}$$

$$\boldsymbol{\varphi}(\mathbf{g}) = \boldsymbol{\varphi}(\operatorname*{argmax}_{\mathbf{q} \in Q^* | \mathbf{g}(\mathbf{q}) = \mathbf{g}} p(\mathbf{q})) \tag{6.17}$$

The software is available as open-source toolkit [Bisani 08].

6.1.5 ps - WFST-based n-gram Model

Phonetisaurus utilizes weighted finite-state transducers for decoding as a representation of a graphone-based n-gram LM trained on data aligned by an advanced M-to-N alignment algorithm [Novak 11]. This alignment is provided by a variant of the EM algorithm [Jiampojamarn & Kondrak$^+$ 07]. For training of the alignment model, the idea is to use a classical forward-backward algorithm within the EM framework. Whereas the grapheme has to be non-empty within each graphone, an empty phoneme is allowed. For decoding, the Viterbi approximation is applied. Additionally, the authors of the applied many-to-many alignment do not use evidence trimming or smoothing techniques. They argue that this algorithm is less complex than the one described in the previous section, but accurate enough.

The n-gram model is trained using the Massachusetts Institute of Technology (MIT) LM toolkit [Bo-June & Glass 08], in which Kneser-Ney discounting with interpolation is used for smoothing. Decoding is done using OpenFST [Allauzen & Riley$^+$ 07] with the following sequence of operations:

$$\text{nBest}(\pi(\text{project\_output}(W \circ M))) \tag{6.18}$$

Here, W denotes the input FSA, which is a graph representation of the input word including grapheme clusters seen in training as alternative paths. M is the n-gram model encoded as FST. The W and M FSTs are composed and a projection onto the output symbols is performed. π denotes the removal of unwanted symbols like the empty token ε or sentence begin and end markers (<s>, </s>).

6.2 Experimental Results

6.2.1 Performance Measurement

For a fair comparison of the various methods, we performed model training and optimization on exactly the same data. The free model parameters are optimized empirically on the development set and the best setting is used for evaluation purposes.

As performance measures, we use PER and WER to assess the quality of the G2P models on text data. The PER is defined as the Levenshtein distance between a hypothesis and a reference pronunciation divided by the number of phonemes in the reference pronunciation. Given a corpus with reference pronunciations φ_1^N and hypotheses $\hat{\varphi}_1^N$, the PER is given as:

Table 6.1 G2P results on the English Celex task for various G2P methods. n denotes the best performing context length.

method	n-gram order	PER/WER[%] training	development	evaluation
ngdt	5	0.8 / 3.8	3.4 / 15.8	3.4 / 15.8
ibm	9	1.4 / 6.4	3.9 / 17.3	3.9 / 17.4
dra	5	0.3 / 1.5	3.5 / 15.5	3.4 / 15.2
seq	8	0.1 / 0.6	2.7 / 11.8	**2.5 / 11.4**
ps	6	1.0 / 6.2	3.6 / 18.0	3.4 / 16.7
ROVER		0.2 / 1.0	2.6 / 12.0	2.5 / 11.6

$$\mathrm{PER}(\hat{\varphi}_1^N, \varphi_1^N) = \frac{\sum_{i=1}^N \mathfrak{L}(\hat{\varphi}_i, \varphi_i)}{|\varphi_1^N|} \tag{6.19}$$

Here, $\mathfrak{L}$ denotes the Levenshtein distance. The WER is defined as the fraction of words containing at least one error:

$$\mathrm{WER}(\hat{\varphi}_1^N, \boldsymbol{\varphi}_1^N) = \frac{\sum_{i=1}^N \delta(\hat{\varphi}_i, \varphi_i)}{N} \tag{6.20}$$

with the δ-function returning zero, if the two strings are identical and one otherwise.

Scoring is done using the NIST sclite scoring toolkit [NIST 95]. In some of the used lexicons, more than one reference pronunciation is given per word. A hypothesis is considered correct if it equals one of the given reference variants. For the ASR experiments, we report the usual word error rates. Note that for both, G2P and ASR, the WER measure is used, but on a different level. Whereas the errors for G2P are measured on phoneme and word level, the corresponding levels in ASR would be word and sentence level.

6.2.2 Results on Text Data

First experiments have been performed on the publicly available, mid-size English The Dutch Centre for Lexical Information (CELEX) lexical database [Baayen & Piepenbrock$^+$ 96]. The corresponding corpus statistics are presented in Section A.3. The results are presented in Table 6.1.

The seq model performs clearly best with a phoneme error rate of 2.5% while all other models are in the same range of about 3.4%, except the IBM model is somewhat worse. It is also interesting to see that the optimal n-gram context size varies between five and nine. For the ngdt approach, the decision tree context lengths are four on source resp. grapheme side and one on target resp. phoneme side. We also applied ROVER system combination [Fiscus 97], but there are no improvements over the best system. One reason might be that the models are too similar since they all rely on graphone-based n-grams and the accuracy of the best performing model is already 25% relatively better than the accuracy of the second best performing model.

Table 6.2 G2P results on the English evaluation set. n denotes the best performing context length for each method. The baseline system will be used for the ASR experiments in the next subsection and is included for reference only.

method	n-gram order	PER/WER[%] training	development	evaluation
baseline	3	6.1 / 26.9	9.2 / 38.8	9.0 / 38.2
ngdt	5	2.2 / 10.0	5.6 / 24.8	5.7 / 24.9
ibm	14	2.6 / 11.8	5.3 / 22.4	5.2 / 22.7
dra	6	0.4 / 2.0	5.6 / 23.7	5.7 / 24.9
seq	9	0.3 / 1.7	4.8 / 21.0	**4.9 / 21.4**
ps	8	0.4 / 1.3	4.8 / 20.6	5.0 / 21.4
ROVER		0.4 / 1.8	4.6 / 20.3	4.6 / 20.6

With respect to alignment restrictions, seq performed best when allowing zero to one symbols on grapheme and phoneme side per graphone, while for ps it was zero to two symbols each.

To get an idea how these methods perform on large state-of-the-art tasks, we built some study sets for five Western-European languages, namely English, German, French, Italian, and Dutch. The corresponding pronunciation dictionary corpora are described in detail in Section A.4, especially Table A.4.

In Table 6.2, the performance on the English data set is presented for the various methods. Here, sequitur achieves the best performance, but the performance gap to phonetisaurus (ps) is only marginal. For this task, ROVER system combination gives a small improvement. Note that this problem is harder than usual, since roughly five times as many characters are allowed within the grapheme input, since the systems should be able to produce a phoneme sequence for more or less any input character sequence, which is important for end user customization. Compared to the baseline G2P system (resulting in a PER of 9.0%), which has been used for the baseline LVCSR experiments presented later in this section, all optimized G2P systems lead to much better phoneme error rates (between 4.9% and 5.7%), whereas the ROVER system combination result improves G2P accuracy by nearly 50% relative.

In Table 6.3, we present the error rates on the evaluation sets for the remaining languages. Overall, seq and ps achieve the best results, whereas system combination does not lead to improvements. It should be noted that the models' performance is already quite good with phoneme error rates ranging from 1.2% to 4.4% across languages and approaches. As already for the English data set, the baseline system could be improved by roughly 50% relative. This system will be used for the ASR experiments in the next section and is included in the tables for reference. Not expected was the high optimal n-gram order for the ibm system, which could already be observed on CELEX in Table 6.1.

Table 6.3 G2P results on the remaining evaluation data sets (PER/WER[%]). The baseline systems will be used for the ASR experiments in the next subsection and are included for reference only.

method	German	French	Italian	Dutch
baseline	6.1 / 47.8	2.7 / 13.3	3.0 / 21.8	2.8 / 18.4
ngdt	4.4 / 36.6	1.5 / 7.2	1.9 / 14.4	1.5 / 9.9
ibm	3.2 / 25.3	1.5 / 7.4	1.8 / 13.0	2.0 / 11.8
dra	3.6 / 27.1	1.9 / 9.1	1.8 / 12.1	1.5 / 9.9
seq	3.3 / 25.5	**1.3 / 6.5**	**1.5 / 10.6**	1.3 / 8.4
ps	**3.1 / 23.7**	**1.3 / 6.4**	1.6 / 11.0	**1.2 / 7.8**
ROVER	3.0 / 23.8	1.3 / 6.3	1.5 / 10.7	1.2 / 7.8

6.2.3 LVCSR Results - Single Best Pronunciation

We applied some of the optimized G2P systems to several study sets to assess the effect on the ASR word error rate. Besides the G2P system used to generate pronunciations for words where no manual transcription is available, the system setup is exactly the same per language. We apply a typical, contemporary two-pass ASR system comprising speaker independent models and adaptation on utterance level. Since we are mostly interested in measuring the difference in recognition quality across various G2P strategies, the exact ASR system setup is not that important to interpret the experimental results.

The corresponding corpora statistics are presented in Section A.4. As one can see, the ratio of words with automatically generated pronunciations is small on the various evaluation sets. Thus, it might be difficult to measure the effect of improved G2P modeling. Additionally, we were particularly interested how the methods studied fared against an established baseline system with accuracies $> 90\%$ (cf. Table 6.2 and 6.3). Since the evaluation corpora are quite large, we always present two decimal places in the result tables (e.g. for English, 0.01 percent in word error rate represents approximately 63 words).

In Table 6.4, we report recognition results for the baseline, the dra and the seq G2P system for all five selected languages. We have chosen the latter two modeling approaches, since they lead to good results on text data and were easy to integrate into the whole LVCSR pipeline. Besides the WER on the complete test sets, we also report the WER on two possibly overlapping subsets. The first one contains utterances with at least one word with automatically generated pronunciation and the second one contains utterances with at least one OOV. As already suspected, the WER does not change much since only few words are affected. Measurable improvements can only be reported for French (from 27.93% WER down to 27.88% WER using dra) and Dutch (from 28.17 % WER down to 28.01% WER also using dra). One should keep in mind that to get these improvements, only 1.66% resp. 1.24% of the pronunciations within the recognition lexicons have potentially been changed by the improved G2P system (cf. Table A.5). If we just look at the error rates for segments which contain at least one pronunciation generated by a G2P model (first number in brackets), the effect of the improved modeling is more prominent,

Table 6.4 ASR recognition results on various LVCSR study sets. The two numbers in brackets refer to the WER considering only segments containing at least one pronunciation generated by a G2P model resp. to the WER on segments containing at least one OOV.

	WER [%] on evaluation set		
	baseline	dra	seq
English	20.32 (28.74 |47.19)	20.32 (**28.26** |47.10)	20.33 (28.38 |47.21)
German	21.30 (30.26 |29.97)	21.31 (30.32 |29.93)	21.29 (**30.17** |29.94)
French	27.93 (36.68 |43.86)	27.88 (**36.36** |43.78)	27.92 (36.63 |43.87)
Italian	24.79 (33.58 |41.23)	24.77 (33.17 |41.18)	24.77 (**33.12** |41.18)
Dutch	28.17 (32.30 |35.95)	28.01 (**32.01** |35.83)	28.08 (32.07 |35.90)

Table 6.5 Cheating experiment: ASR recognition results on various LVCSR study sets, where all OOVs from the evaluation set have been added to the recognition vocabulary. The two numbers in brackets refer to the WER considering only segments containing at least one pronunciation generated by a G2P model resp. to the WER on segments containing at least one OOV.

	WER [%] on evaluation set incl. OOVs in recognition vocabulary		
	baseline	dra	seq
English	19.84 (28.44 |36.40)	**19.78** (27.94 |34.87)	**19.78** (28.08 |34.56)
German	19.69 (28.85 |24.80)	19.66 (28.84 |24.71)	**19.58** (28.59 |24.49)
French	27.31 (36.22 |38.89)	**27.24** (35.92 |38.52)	**27.24** (36.06 |38.35)
Italian	23.29 (32.52 |33.64)	**23.22** (32.06 |33.14)	23.28 (31.99 |33.43)
Dutch	27.06 (31.58 |32.00)	26.89 (31.20 |31.68)	**26.88** (31.27 |31.55)

e.g. for English it is around 2% relative using the dra system (from 28.74% WER down to 28.26%). Concerning the segments containing OOVs, there is not much difference between the improved G2P models and the baseline system. This was to be expected, since OOVs always lead to errors which can not be recovered by merely changing pronunciations in the recognition lexicon.

Since our goal was to measure the quality of the G2P systems, we did a cheating experiment and added all OOVs to the ASR vocabulary and used the respective G2P system to generate their pronunciations. Thus, the number of pronunciations in the recognition lexicon which has been generated automatically could be enlarge by the OOV ratio (cf. Table A.5). These results are presented in Table 6.5. Now, as expected, the effect of improved G2P modeling is stronger, especially when looking at the OOV segments only. For example, for English, when the baseline G2P model is used, the WER on OOV segments is 36.40% and drops to 34.56% when the optimized seq model is used instead, which is an improvement of roughly 5% relative. Overall, both the dra and seq method lead to improved LVCSR results, but there is no clear best method, i.e. the best method varies with the language.

Table 6.6 Comparison of ASR recognition results on various LVCSR study sets. Here, the Sequitur model has been used with first-best pronunciation only as well as with an n-best strategy. The two numbers in brackets refer to the WER considering only segments containing at least one pronunciation generated by a G2P model resp. to the WER on segments containing at least one OOV.

	WER [%] on evaluation set	
	seq	seq-n-best
English	20.33 (28.38 |47.21)	20.29 (**26.95** |46.95)
German	21.29 (30.17 |29.94)	21.40 (**29.62** |30.09)
French	27.92 (36.63 |43.87)	27.87 (**36.05** |43.70)
Italian	24.77 (33.17 |41.18)	24.77 (**33.12** |41.18)
Dutch	28.08 (32.07 |35.90)	28.05 (**31.71** |35.93)

Table 6.7 Cheating experiment: Comparison of ASR recognition results on various LVCSR study sets. Here, the Sequitur model has been used with first-best pronunciation only as well as with an n-best strategy. All OOVs from the evaluation set have been added to the recognition vocabulary using a G2P model. The two numbers in brackets refer to the WER considering only segments containing at least one pronunciation generated by a G2P model resp. to the WER on segments containing at least one OOV.

	WER [%] on evaluation set incl. OOVs in recognition vocabulary	
	seq	seq-n-best
English	19.78 (28.08 |34.56)	**19.68** (26.69 |32.36)
German	19.58 (28.59 |24.49)	**19.53** (28.02 |24.03)
French	27.24 (36.06 |38.35)	**27.16** (35.56 |37.77)
Italian	23.28 (31.99 |33.43)	**23.11** (31.17 |32.42)
Dutch	26.88 (31.27 |31.55)	26.88 (30.83 |31.18)

6.2.4 LVCSR Results - n-Best Pronunciations

We did one run of experiments where we added n-best pronunciation variants for all words where no manual pronunciation is available instead of single-best, presented in Table 6.6. Here, we used the seq model and generated up to three pronunciations, depending on the overall posterior probability mass of the generated variants, which has been thresholded to < 0.75. But the overall (non-cheating) error rates do not necessarily improve, since more confusion may be introduced (e.g. for German). If we only consider segments containing automatically generated pronunciations, there is a small but consistent improvement across languages if n-best pronunciations are used.

When looking at the cheating experiments presented in Table 6.7, the variants seem to help for all languages except Dutch where there is at least no degradation in performance. One reason for Dutch beeing an exception might be that G2P for this language is mostly rule-based and more or less defined by the spelling.

6.3 Conclusions

We have presented a comparison of various state-of-the-art G2P models on large tasks on both G2P accuracy and their effect on ASR performance within contemporary LVCSR systems on challenging tasks. The Sequitur and Phonetisaurus tools seem to outperform the other tested methods. With a cheating experiment where OOVs have been added to the recognition lexicon, it could be shown that improved G2P modeling can be measured within ASR systems even over a highly competitive baseline. Additionally, using n-best pronunciations seems to help for most languages. In any case, improving G2P is always beneficial for end user customization.

Chapter 7

CRFs for G2P

In Chapter 4, it has been shown that discriminative methods based on CRFs perform very well on NLU tasks. In this chapter, CRFs as introduced in Section 1.4.3 are now applied to the G2P problem. Since there are some fundamental differences in the two tasks, a number of issues have to be addressed.

Due to the very different nature of the G2P task, the used features have to be revised. In Section 7.1, we introduce the features which have been found to be useful.

Since the number of training samples for G2P, i.e. the size of pronunciation dictionaries, is usually much larger than the number of training sentences for NLU tasks (cp. e.g. Tables A.4 and A.1), the number of (context dependent) observed features is much larger, too. Thus, there is a need for techniques to reduce the amount of features resp. to speed-up the training process. In Section 7.2, we present a way to substitute the costly bigram feature for a monolingual LM on target side in search as a short digression for a task with a given manual alignment. In the following Section 7.3, the elastic net (EN) is introduced within the RProp optimization algorithm as well as feature cut-offs.

As already discussed, to apply CRFs, a one-to-one alignment between source and target side is necessary to describe context dependent feature functions. Since this is not the case for most real-life G2P tasks (in contrast to NLU), we will introduce the alignment within CRFs, called HCRFs, in Chapter 8. Since the resulting training criterion is not convex anymore, a good initialization of the model is crucial, as presented in Section 8.6 and [Guta 11]. Additional features which are helpful for the letter-to-sound conversion task will also be presented. To get an idea of the effect of using CRFs for LVCSR, we present extensive experimental results on the English QUAERO 2010 task in Chapter 9.

In [Lavergne & Cappé$^+$ 10, Sokolovska & Lavergne$^+$ 10], CRFs have already been applied to the G2P task, but only when a manual alignment between source and target side is provided.

7.1 Features for G2P

In Section 1.4.1, we have introduced a couple of task-dependent feature functions which have worked well for the NLU task but have to be revised for the G2P task. Since our smallest unitis are now letters in contrast to words within NLU, prefix, suffix or capitalization feature are not meaningful. Thus, those features will not be used anymore. Instead, the source-n-gram features, which did not work well for the NLU task, will be used, since it is meaningful to model n-gram dependencies on grapheme side. Since in contrast to words, the number of

...	$g_{a_{n-3}}$	$g_{a_{n-2}}$	$g_{a_{n-1}}$	g_{a_n}	$g_{a_{n+1}}$	$g_{a_{n+2}}$	...
...	φ_{n-3}	φ_{n-2}	φ_{n-1}	φ_n	x	x	...

Figure 7.1 General example for a source-n-gram feature.

t	h	o	u	g	h	t
T_s	T_c	$O{:}_s$	$O{:}_c$	$O{:}_c$	$O{:}_c$	t_s

Figure 7.2 English example for a source-n-gram feature.

graphemes per language is usually very limited, at least for the analyzed Western-European languages, patterns are to be expected to be seen in the training material leading to improved model performance. An example for source-n-gram features is given in Figure 7.1.

In contrast to the already introduced lexical feature, where exactly one token on source side is considered in conjunction with the currently observed token on target side, a source-n-gram feature can be a more or less arbitrary n-gram on source side. Since there usually is no one-to-one alignment given in G2P tasks, an alignment between graphemes and phonemes has to be calculated or given. The grapheme aligned to the phoneme at position n is represented by the subscript a_n in the figure. One English example for a source-n-gram feature is given in Figure 7.2. Here, for reasons of clarity, the alignment is represented as a one-to-one alignment using the BIO scheme as introduced in Section 1.2.1. Examples for other possible alignments within the G2P task are shown in Figure 8.1. Note that there might be no restrictions on the kind of alignment (i.e. a real many-to-many alignment) as presented in Section 8.6 for HCRFs. Additionally, empty phonemes and graphemes might be introduced.

Basically, lexical features, bigram features, and the source-n-gram features will be used for all of the following experiments. It should be noted that the number of features will increase heavily by including the source-n-gram features and some mechanisms have to be used to deal with this issue.

7.2 LM in CRF Search

One of the most powerful features of CRFs is the context feature on target side, the so-called bigram feature. It is computationally expensive, since the complexity of the model correlates with the context length, as already described in Section 1.4.3. Since longer context may lead to even better results but the complexity forbids the direct integration (at least for real-life tasks) one solution could be to integrate a classical LM into the CRF search process. This LM could easily be calculated beforehand and outside of the CRF framework by considering only the data given on the target side of the training corpus. It is used as an additional knowledge source:

$$\hat{\varphi}_1^N = \underset{\varphi_1^N}{\operatorname{argmax}} \left\{ \underbrace{exp(\sum_{m=1}^{M} \lambda_m \cdot h_m(\varphi_{n-1}, \varphi_n, g_1^N))^{1-\alpha}}_{\text{original CRF model}} \cdot \underbrace{p_{LM}^{\alpha}(\varphi_1^N)}_{\text{additional ARPA LM}} \right\} \tag{7.1}$$

The LM is weighted using α.

On the one hand, it might be possible to get already good performance when omitting the bigram feature and just rely on the LM to represent the context information on phoneme side. This might lead to a speed-up of the CRF training process as well as reduce the needed memory for the model. Note that a CRF without bigram features is equivalent to an MEMM without bigram features. On the other hand, a longer context than bigrams may be beneficial even additionally to the bigram feature. Though it would be theoretically possible to model longer context dependencies on phoneme side directly within CRFs, the computational complexity would be very high (cf. [Lehnen & Hahn$^+$ 11b] for results on the NETtalk 15k data base, where at least trigrams are used in training). We would not have any problems integrating an n-gram LM with a much higher order.

The SRI LM Toolkit has been used to train the additional standard ARPA format language models [Stolcke 02]. Experimental results are reported in the next section.

7.2.1 Experimental Results: LM Perplexity

For our experiments, we have chosen the English NETtalk 15k corpus. Details about this G2P dataset are presented in Section A.2. Since a large number of experiments with a detailed analysis had to be performed, such a small corpus with manual aligned graphemes and phonemes is well suited.

Note that there are 50 different phonemes on target side, which are represented by 100 tags within the CRF, since "begin" and "continue" tags are used (cf. BIO-scheme as described in Section 1.2.1) . Additionally, an empty "NULL" phoneme is introduced by the manual alignment, which results in an overall vocabulary size of 101. Additional word start and end tokens are introduced by the SRI LM toolkit. Overall, the LM has been trained on 13,935 phoneme sequences with a total of 102,493 running phoneme tags. The statistics and perplexities of the resulting LMs in various orders are given in Table 7.1.

The effect of a language model integrated into CRF search has been measured in the following way: first, standard ARPA LMs have been trained on the phoneme side of the training corpus for the orders two up to seven. The lowest perplexity on the development set is obtained for an n-gram order of five with 8.3 (cf. Table 7.1). These results indicate that a context larger than 4 might not help to improve G2P systems. Second, the LMs have been applied to various CRF G2P systems, as presented in the next section.

7.2.2 Experimental Results: G2P

To evaluate the G2P systems, we used the typical error measures PER as well as WER.

We first optimized a baseline system on the corpus. Therefore, we checked a number of different features and came up with the following setup leading to our best result: lexical features

Table 7.1 Statistics and perplexities for various language models trained on the NETtalk 15k corpus. The LM of order n always includes all n-grams of orders smaller than n.

order n	# n-grams of order n	perplexity on dev set
1	103	40.7
2	1.989	12.3
3	7.142	9.5
4	13.024	8.4
5	12.025	8.3
6	7.762	8.4
7	4.438	8.5

in a window of $[-4,\ldots,4]$ around the current word, i.e. at nine positions, the bigram feature and combined or source n-gram features. The latter features are composed of all monotone and overlapping combinations of lexical features of lengths two up to six. See Section 1.4.1 in conjunction with Section 7.1 for a detailed description of the various features used within CRFs for G2P. Additionally, the margin extension to the training criterion (cf. Section 5.2.2) is used for all reported experiments. Since we wanted to measure the effect of the LM on phoneme side versus the bigram feature, we additionally performed some experiments where we added a unigram feature instead of a bigram feature in CRF training. The respective features are denoted as h_0 and h_1 in Table 7.2.

We wanted to separate the effect of the various kinds of features and the language model. Thus, we trained six different CRF systems, each incorporating different features. Table 7.2 gives an overview of the selected features and the respective experimental results. To each of the systems, the language models on phoneme side of order two up until seven have been combined. Therefore, for each experiment, the interpolation weight α had to be adjusted. We did this by grid search. The order and weighting factor of the best performing LM are also presented in Table 7.2. The results are grouped according to the used features. Each experiment is reported with and without the additional LM in search. To improve result discussions, system numbers have been added to the various experiments within the table.

If we have a look at the experiments where no combined features are incorporated, we can see that the LM can improve the system, even if bigram dependencies are considered within the CRF (third of the six experiments within the table, comparing systems 5 and 6). The improvement is also more or less independent of the presence of a unigram feature (second experiment; systems 3 and 4) or no context feature on phoneme side (first experiment; systems 1 and 2). As soon as combined features are incorporated, the quality of the model greatly improves (even more than with the bigram features alone). Here, the additional LM can not improve the best performing system significantly (last experiment; systems 11 and 12), but it can in someway compensate for the bigram feature, since the result of the model with only the unigram feature can be improved with a 4-gram LM to give the same performance as with the bigram feature (compare the improvement of adding an LM to system 9, resulting in system 10,

Table 7.2 Results for language models on phoneme side integrated into CRF search. Additionally to the CRF features, the best performing LM w.r.t. this feature set (n-gram order as well as the interpolation weight α) is given. Tuning of LM for Experiment "*" is documented in Figure 7.3. h_0 and h_1 represent the unigram and bigram feature respectively, lexical features are $(t_n = t', s_{n+\delta} = s')$, and source n-grams are combinations of successional lexical features.

feature set			system number	LM	PER[%] dev	PER[%] eva	WER[%] dev	WER[%] eva	LM order / α
source lexicals			1		14.6	14.6	59.7	57.5	–
			2	✓	11.5	11.8	47.2	46.7	7 / 0.35
		$+h_0$	3		14.8	14.6	60.2	57.6	–
			4	✓	11.7	11.9	47.0	47.0	7 / 0.40
		$+h_1$	5		12.3	12.2	51.9	49.5	–
			6	✓	11.1	11.1	45.5	44.8	6 / 0.30
	+ source n-grams		7		8.0	8.3	35.4	35.9	–
			8	✓	7.4	8.3	33.0	35.8	6 / 0.20
		$+h_0$	9*		8.0	8.3	35.0	36.0	–
			10	✓	7.4	7.9	32.6	34.5	4 / 0.25
		$+h_1$	11		7.4	7.9	32.3	34.2	–
			12	✓	7.3	7.8	32.1	33.5	5 / 0.10

with the results for system 11).

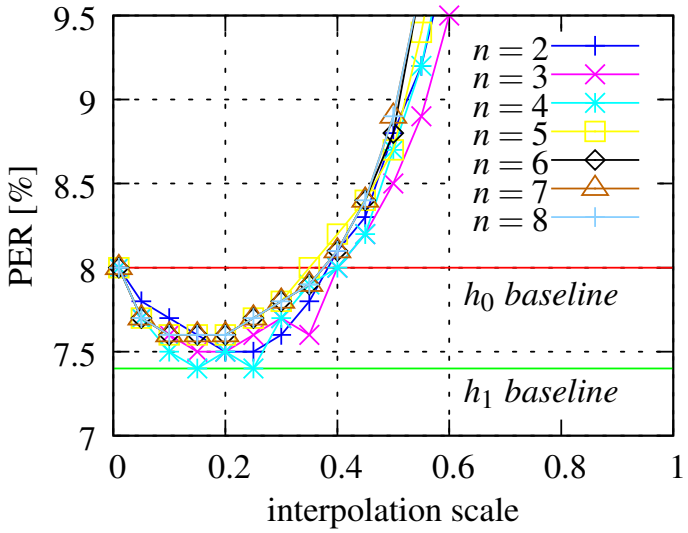
This is also true, if the EN is applied (cf. Section 7.3). Thus, it would be possible to omit the bigram feature and decrease training time (to about 10% of the original time) and memory requirements greatly at the price of an increased time for search respectively testing due to application of the additional LM. Note that the results in Table 7.2 are roughly 6% relatively better than those reported in [Bisani & Ney 08], the most current publication using exactly this corpus. This can to some degree be explained with the fact that discriminative models usually work better than generative ones if little training data is available.

In Figure 7.3, the tuning of the language model for the system marked with a "*" in Table 7.2 is presented. As can be seen, even with an LM of order two, it is possible to get close to the performance of the CRF bigram feature system. The optimal interpolation weight α is always between 0.1 and 0.3 for the various n-gram orders. Thus, the major weight is still on the CRF features, which was to be expected. As can be seen in Table 7.2, the more powerful the CRF feature set-up, the lower is the optimal interpolation weight, i.e. $\alpha = 0.1$ for the 5-gram LM and thus the best system.

7.2.3 Conclusion

Although we could show that it is possible to substitute the costly bigram feature with a standard LM trained on phoneme sequences interpolated in search, it did not lead to improvements, even for larger contexts than two. Thus, we will stick to the conventional CRF bigram feature for most of the remaining experiments reported in this thesis, since we then have a closed mathematical framework and optimization process for the model training. But it should be

Figure 7.3 Effect of the LM on the performance (PER on the development set) of the CRF system ("*" in Table 7.2) for various interpolation scales. The h_0 baseline feature set here includes lexical and combined features as well as a unigram feature; h_1 baseline indicates the same system with bigram features instead, the overall best system.



noted that it might still be interesting for tasks like SMT where the bigram feature might be even more expensive.

7.3 Elastic Net for RProp

7.3.1 Introduction

Since usually overlapping features are used within CRFs, the total number of features λ_1^M on tasks like G2P can easily lead to feature sets with hundreds of millions of features. A weight for each of these features has to be learned. The resulting non-linear optimization problem is convex (i.e. there is a global optimum; see some notes about convexity of CRFs in e.g. [Sutton & McCallum 10]) and usually tackled via a gradient-based hill-climbing algorithm like Broyden-Fletcher-Goldfarb-Shanno (BFGS) or limited memory BFGS (LBFGS). Such so-called Quasi-Newton methods need to keep the feature weights, the gradient of the conditional log-likelihood as well as at least a sparse approximation to the Hessian in memory. There are basically two possibilities to make this task feasible: on the one hand, a pre-selection of useful features would reduce the computational complexity. Additionally, since the training material is limited, most of the features are rarely observed and can thus not be trained reliably. A good feature (pre-)selection might even improve the overall quality of the model. On the other hand, a faster or simpler optimization algorithm would reduce the immense computational and memory requirements and would thus allow for more features to be included in training.

In [Zou & Hastie 05], the so-called elastic net (EN) is introduced. It is a combination of L2

and L1 regularization, with 2- and 1-Norm $||\cdot||_{1/2}$:

$$r(\lambda_1^M) = c_2||\lambda_1^M||_2^2 + c_1||\lambda_1^M||_1 \tag{7.2}$$

$$= c_2 \sum_{m=1}^{M} \lambda_m^2 + c_1 \sum_{m=1}^{M} |\lambda_m| \tag{7.3}$$

Compared to the regularization as introduced in the original training criterion in Equation 1.38, an L1 regularization has been added. This leads to a modified training criterion over a training set $\{\{\bar{t}_1^N\}_k, \{s_1^N\}_k\}_{k=1}^K$:

$$L = \log p(\lambda_1^M) + \sum_{s=1}^{S} \log p_{\lambda_1^M}(\{\bar{t}_1^N\}_s | \{w_1^N\}_s) - r(\lambda_1^M) \tag{7.4}$$

The advantage of introducing an additional L1 regularization according to the authors is the implicit feature selection due to the fact that such a regularization leads to sparse parameter vectors which contain by definition many zeros. For details on the optimization theory behind the EN, the reader is referred to [Tibshirani 94,Zou & Hastie 05]. Concerning CRFs, the authors of [Lavergne & Cappé$^+$ 10] present an integration of the EN within orthant-wise quasi-Newton (OWL-QN), stochastic gradient descent (SGD) and block coordinate descent. Although the reported decrease in training time is promising, further reductions are still desirable.

7.3.2 The Method

In [Hahn & Lehnen$^+$ 11], an extension to the popular RProp algorithm is presented, which introduces the idea of the original EN. The RProp algorithm as originally proposed in [Riedmiller & Braun 93] is shown in Figure 7.4.

The idea of the RProp algorithm is to just rely on the sign of the gradient to update the weights λ_1^M. The stepsize s_m is adjusted per weight λ_m and calculated in two steps. In the first step, the updated stepsize is calculated based on a possible change in the gradient (the if-statement in the algorithm). In the second step, the weight is adjusted according to the updated stepsize. There are three cases to be distinguished: first, the sign of the gradient did not change between iteration $i-1$ and i. In this case, the updated stepsize $s_{m,i+1}$ for the next iteration is increased by a factor $s^+ > 1$ and the weight is updated in the direction of the gradient. Second, the sign of the gradient did change between the iterations $i-1$ and i. In this case, the algorithm overstepped the optimum. Thus, the stepsize for the next iteration is decreased by a factor $0 < s^- < 1$, the weight is set back to the weight of the previous iteration (i.e. it is not updated) and the current gradient is set to zero. Thus, the next step will be executed in the same direction as in the previous iteration, but with a smaller stepsize. Third, the product of the gradient of iteration $i-1$ and i equals zero. In this case, the current stepsize is kept, i.e. not updated. The weight is updated in the direction of the gradient from the current iteration.

As already discussed, there are usually many features which are non-discriminative or unimportant and will thus get a zero weight eventually. Thus, since we use an iterative optimization method, some lambdas will become zero after an (possibly) infinite number of RProp iterations.

Input: previous and current lambdas $\{\lambda_1^M\}_{i-1}$, $\{\lambda_1^M\}_i$
current step sizes $\{s_1^M\}_i$,
previous and current gradient of the objective function $\{\nabla_{\lambda_1^M} L\}_{i-1}$, $\{\nabla_{\lambda_1^M} L\}_i$
Output: updated lambdas $\{\lambda_1^M\}_{i+1}$
updated step sizes $\{s_1^M\}_{i+1}$
repeat
for $m \in 1,\ldots,M$:
if $\{\frac{\partial L}{\partial \lambda_m}\}_{i-1} \cdot \{\frac{\partial L}{\partial \lambda_m}\}_i > 0$:
$s_{m,i+1} = \min(s^+ \cdot s_{m,i}, s_{max})$
$\lambda_{m,i+1} = \lambda_{m,i} - sign(\{\frac{\partial L}{\partial \lambda_m}\}_i) \cdot s_{m,i+1}$
else if $\{\frac{\partial L}{\partial \lambda_m}\}_{i-1} \cdot \{\frac{\partial L}{\partial \lambda_m}\}_i < 0$:
$s_{m,i+1} = \max(s^- \cdot s_{m,i}, s_{min})$
$\lambda_{m,i+1} = \lambda_{m,i-1}$
$\{\frac{\partial L}{\partial \lambda_m}\}_i = 0$
else if $\{\frac{\partial L}{\partial \lambda_m}\}_{i-1} \cdot \{\frac{\partial L}{\partial \lambda_m}\}_i == 0$:
$s_{m,i+1} = s_{m,i}$
$\lambda_{m,i+1} = \lambda_{m,i} - sign(\{\frac{\partial L}{\partial \lambda_m}\}_i) \cdot s_{m,i+1}$
until converged

Figure 7.4 Rprop Algorithm as proposed in [Riedmiller & Braun 93]. The stepsize related variables s^+, s^-, s_{max}, s_{min} are freely adjustable and typically set to $s^+ = 1.2$, $s^- = 0.5$, $s_{max} = 50$, $s_{min} = 10^{-6}$.

The idea now is to derive a method to detect these lambdas early and strictly fix them to zero, i.e. to not calculate any new gradients for those weights in later iterations. To minimize computational overhead, this should be done with only the information which is already available by the RProp algorithm. In [Hahn & Lehnen$^+$ 11], the following three equations have been derived to achieve this task:

$$0 \geq \lambda_{m,i} \cdot \lambda_{m,i-1} \tag{7.5}$$

$$0 \geq \left\{\frac{\partial L}{\partial \lambda_m}\right\}_{i-1} \cdot \left\{\frac{\partial L}{\partial \lambda_m}\right\}_i \tag{7.6}$$

$$c_1^2 > (\left\{\frac{\partial L}{\partial \lambda_m}\right\}_i + c_1 sign(\lambda_m))^2 \tag{7.7}$$

The first Equation 7.5 ensures that the sign of the weight changed from iteration $i-1$ to iteration i which means that we passed the zero value. Additionally, the second Equation 7.6 ensures that the gradient also changed the sign, i.e. the optimum has been overstepped in the current iteration. Thus, we know that the optimal weight might be close to zero. Since we do not know the slope of the gradient, we still need a way to define the proximity. This is achieved with the third Equation 7.7. Here, the value chosen for the L1 regularization c_1 is added to the

Table 7.3 Effect of using elastic net (EN) to reduce the number of features on the grapheme-to-phoneme conversion performance on the English NETtalk 15k corpus. Here, EN uses $c_1 = 2^{-4}$, $c_2 = 2^{-3}$.

	#features	PER[%]		WER[%]	
setup		Dev	Eva	Dev	Eva
full model	54.603.236	7.7	7.9	33.8	34.2
+margin	54.597.879	7.4	7.9	32.3	34.2
elastic net (EN)	322.248	7.5	8.0	33.4	34.3

current gradient. Thus, the L1 regularization is roughly cancelled and we get an approximation to L_0, the objective function without regularization. We can now check if the resulting value is within the bounds of c_1. If this is the case, the current value of λ_m is close to zero and will eventually snap in. If all three equations are true, we can directly set $\lambda_{m,i}$ to zero and skip the if-clauses in the RProp algorithm for the respective feature function. At $\lambda_{m,i} == 0$, it is sufficient to evaluate Equation 7.7. The percentage of feature weights set to zero can now implicitly be guided by the value chosen for the L1 regularization c_1.

Although it is still necessary to calculate the gradient for all weights λ_m, the features which have been set to zero using the three equations presented above do rarely change between iterations, a fact which is currently not exploited in our system. Thus, the sparsity of the data structures as well as skipping the if-clauses in the RProp algorithm leads to improvements in memory and computation time. Experimental results are presented in the next section.

7.3.3 Experimental Results

First experiments have been performed on the English NETtalk 15k corpus (see Section A.2 for a detailed corpus description). First, we optimized a system on the NETtalk corpus, as described in Section 7.2.2. We trained this system with and without the margin extension to the training criterion and report experimental results with and without EN in Table 7.3.

The margin extension does not lead to improvements in PER on the evaluation set. Using EN, it was possible to reduce the set of active features from 55M down to roughly 322K, which corresponds to a reduction of more than 99% down to less than 1% of the features. The performance on the evaluation set does not deteriorate.

In a second experiment, we wanted to analyze the effect of the EN on interpolating an LM within search as presented in Section 7.2.2. The respective results are given in Table 7.4.

The elastic net does not seem to have a large effect on the performance of the LM. The PER on the evaluation corpus using a unigram context feature with or without EN does not change significantly, whereas the difference in WER in negligible (34.2% versus 34.5%).

7.3.4 Conclusion

In this section, we have introduced the EN framework within the RProp algorithm. Thus, we were able to reduce the number of active features to below 1% of the features of the original, full model without loss of performance. For the following experiments, EN will usually be

Table 7.4 Results for language models integrated into CRF search and interaction with elastic net.Additionally, the best LM n-gram order as well as the interpolation weight α is given. h_0 and h_1 represent the unigram and bigram feature respectively. The same lexical as well as source-n-gram features as in Table 7.2 are used. *EN* marks the experiment using elastic net in combination with the language model.

context feature		LM	PER[%] dev	PER[%] eva	WER[%] dev	WER[%] eva	LM order / α
			8.0	8.3	35.4	35.9	–
		✓	7.4	8.3	33.0	35.8	6 / 0.20
$+h_0$			8.0	8.3	35.0	36.0	–
		✓	7.4	7.9	32.6	34.5	4 / 0.25
	EN		7.9	8.4	35.7	36.5	–
		✓	7.5	7.9	33.8	34.2	4 / 0.25
$+h_1$			7.4	7.9	32.3	34.2	–
		✓	7.3	7.8	32.1	33.5	5 / 0.10

used since the difficulty and size of the tasks is increasing and training full models would be too time and memory consuming.

Chapter 8

HCRFs for G2P

In contrast to NLU corpora, there is usually no alignment provided within the training data for G2P models, i.e. within pronunciation dictionaries. Since an alignment is needed to well-define feature functions for CRFs, and since we can assume that there is a "natural" alignment hidden in the training data, we need to find methods to learn this alignment. We also assume that learning an alignment directly with CRFs can help to predict the phoneme sequence. Naturally, the alignment quality affects the error rate of the G2P model.

In this chapter, we will first analyze the G2P alignment problem in general in Section 8.1, followed by three different approaches to tackle this task. The simplest way to get alignment information needed for CRF training would be to use an external model as presented in Section 8.2, i.e. not based on CRFs, to precompute an alignment. This alignment is then kept fix for the CRF training process. There are some disadvantages in this approach though. On the one hand, it would be desirable to be independent from an additional model since this will lead to error propagation and it will be impossible to recover from these errors in the CRF training. On the other hand, the alignments are optimized for certain tasks/tools and might be suboptimal for CRFs.

A second approach which does not rely on an external model would be an EM-like integration of the alignment into CRF training and is described in Section 8.3. The third approach would be a real integration of the alignment within CRFs in form of a hidden variable. Since our software is designed for NLU tasks, where many-to-one alignments suffice, we present an approach which results in restricted HCRFs. The respective model is described in Section 8.4. Comparisons of all these methods will be presented in Section 8.5.

Since one-to-one and many-to-one alignment usually do not suffice for the G2P task in general, we introduce a method capable of generating many-to-many alignments in Section 8.6 leading to HCRFs. This method will then be applied to a real-life, large scale LVCSR task in Chapter 9.

8.1 Alignment Constraints and General System Setup

Since we are only interested in modelling the alignment between graphemes and phonemes for real linguistic pronunciation data, we can introduce some restrictions to the alignment process which excludes impossible alignments w.r.t. pronunciation relation. Additionally, this will lead to a smaller search space for the alignment. Let g_1^L and φ_1^N be a corresponding pair of word

and pronunciation. $a_l = n$ represents the alignment between a grapheme and a phoneme sequence. This mapping returns the position n of φ_n with which g_l is aligned. The following three constraints are used to restrict the possible alignments:

$$\forall l_1 < l_2 : \quad a_{l_1} \leq a_{l_2} \tag{8.1}$$

$$\forall n \exists l : \quad a_l = n \tag{8.2}$$

$$\forall l, n, n' : \quad n \neq n' \curvearrowright a_l \neq n \vee a_l \neq n' \tag{8.3}$$

The first constraint in Equation 8.1 enforces the alignment to be monotonous. This is reasonable, since the graphemes are pronounced in the order of writing. The second constraint in Equation 8.2 ensures that each grapheme is aligned to a phoneme. This representation constraint is meaningful, since there is always one grapheme or a group of graphemes triggering a certain sound resp. phoneme. The last constraint in Equation 8.3 forces all alignments to be M-to-one alignments.

Since CRFs enforce g_1^L and φ_1^N to have the same length, i.e. $L \stackrel{!}{=} N$, we need to introduce the BIO scheme on phoneme side (cf. Section 1.2.1). Thus, we are able to model many-to-many alignments with one constraint: the number of graphemes has to be at most the number of phonemes, i.e. $L \leq N$. Since there are rarely words with more phonemes than graphemes, this restriction is acceptable for the time being. More flexible alignments will be presented in Section 8.6. In Figure 8.1, some examples of such M-to-one alignments extended to one-to-one alignments are presented using SAMPA notation for the phonemes.

Note that in the corpora used for the experiments, a notation deviating from SAMPA notation is used for monophthongs and diphthongs as described in Section A.3. It should be also noted that the square brackets in the phonetic notation are part of SAMPA notation and not of the phonemes. These brackets denote a phonetic transcript in contrast to regular text.

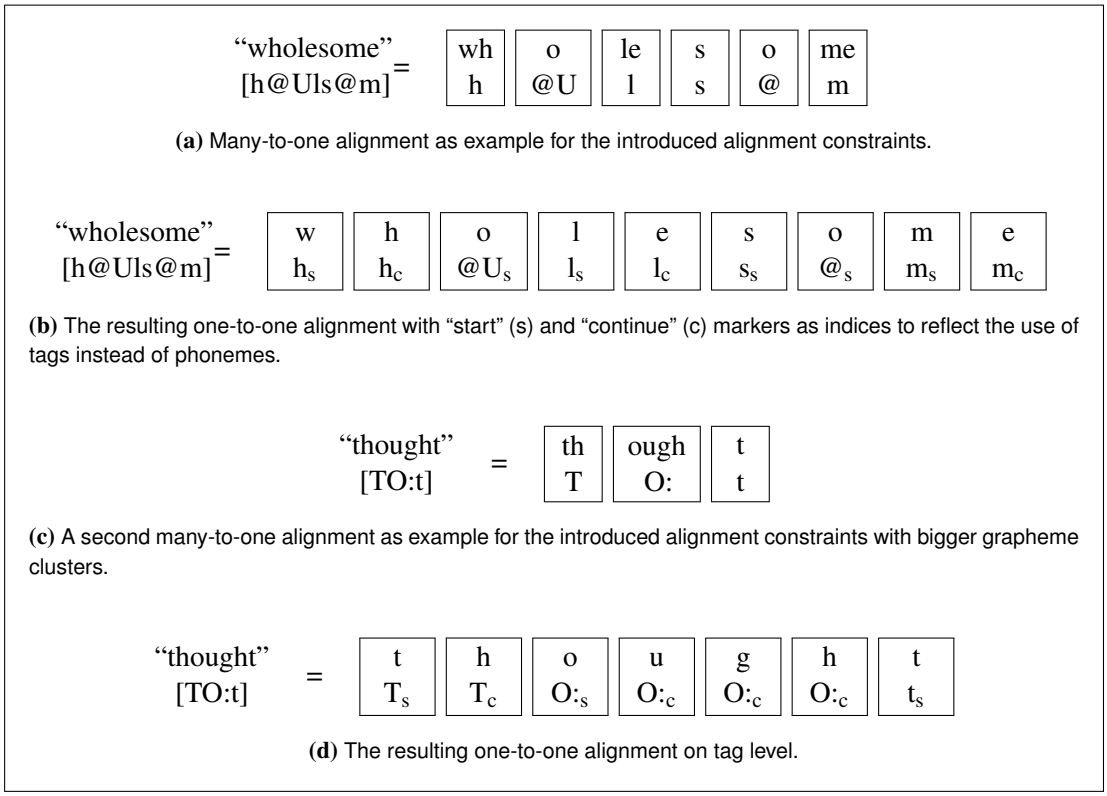
Another important property of using the BIO scheme is that the resulting alignment with the "start" and "continue" tags can be uniquely mapped back to phoneme sequences, i.e.:

$$Pr(\varphi_1^N, a_1^L \mid g_1^L) = Pr(t_1^L \mid g_1^L) \tag{8.4}$$

Here, t_1^L denotes the phoneme-tag sequence. The general task now is to find a model which approximates the true probability of the (unaligned) phoneme sequence given the corresponding grapheme sequence:

$$Pr(\varphi_1^N \mid g_1^L) = \sum_{a_1^L} Pr(\varphi_1^N, a_1^L \mid g_1^L) \tag{8.5}$$

The three different approaches will be introduced in the following section. For all of these approaches, the used feature functions are lexical features, bigram features and source-n-gram features. The overall training methodology is also very similar and based on the forward-backward algorithm using dynamic programming. For the optimization of the feature weights, we use the RProp algorithm as introduced in Section 7.3.2.



(a) Many-to-one alignment as example for the introduced alignment constraints.

(b) The resulting one-to-one alignment with "start" (s) and "continue" (c) markers as indices to reflect the use of tags instead of phonemes.

(c) A second many-to-one alignment as example for the introduced alignment constraints with bigger grapheme clusters.

(d) The resulting one-to-one alignment on tag level.

Figure 8.1 Examples of extending a many-to-one alignment to a one-to-one alignment using the BIO scheme. The SAMPA notation has been used for the phoneme sequences. The boxes indicate graphones.

8.2 External Alignment Model

A straight-forward way to obtain alignment information for CRF training is to use some external alignment model resp. its first-best alignment $\hat{a}_1^L$:

$$Pr(\varphi_1^N \mid g_1^L) = \sum_{a_1^L} Pr(\varphi_1^N, a_1^L \mid g_1^L) \tag{8.6}$$

$$\approx \max_{a_1^L} \left\{ Pr(\varphi_1^N, a_1^L \mid g_1^L) \right\} \tag{8.7}$$

$$\approx Pr(\varphi_1^N, \hat{a}_1^L \mid g_1^L) \tag{8.8}$$

with

$$\hat{a}_1^L = \operatorname*{argmax}_{a_1^L} \left\{ \underbrace{Pr(a_1^L \mid g_1^L)}_{\text{external model}} \right\} \tag{8.9}$$

Using the resulting alignment $\hat{a}_1^L$, the phoneme sequence φ_1^N and the grapheme sequence g_1^L, it is easily possible using the BIO scheme to derive a phoneme tag sequence $\hat{t}_1^L$ leading to a one-to-one alignment between graphemes and phoneme tags, which will be the target sequence for the training of the CRF model $p_{\Lambda,CRF}(\hat{t}_1^L \mid g_1^L)$. The phoneme tag sequence will be converted back to phonemes after decoding prior to evaluating the system. Note that the CRF training is still a convex optimization problem using the external alignment.

Although this is only an approximation to the desired integrated alignment, it is an easy way to get aligned training material needed for CRFs. In this section, various external methods are presented and compared. We will start with a linear segmentation, a very rough approximation, in Section 8.2.1 followed by a machine translation approach based upon giza++ [Och & Ney 00a, Och & Ney 00b] in Section 8.2.2. We also used the joint-n-gram approach as described in Section 6.1.4. Additionally, we will also compare these three approaches with a manual alignment on the NETtalk corpus, which can also be regarded as a type of external alignment.

8.2.1 Alignment from Linear Segmentation

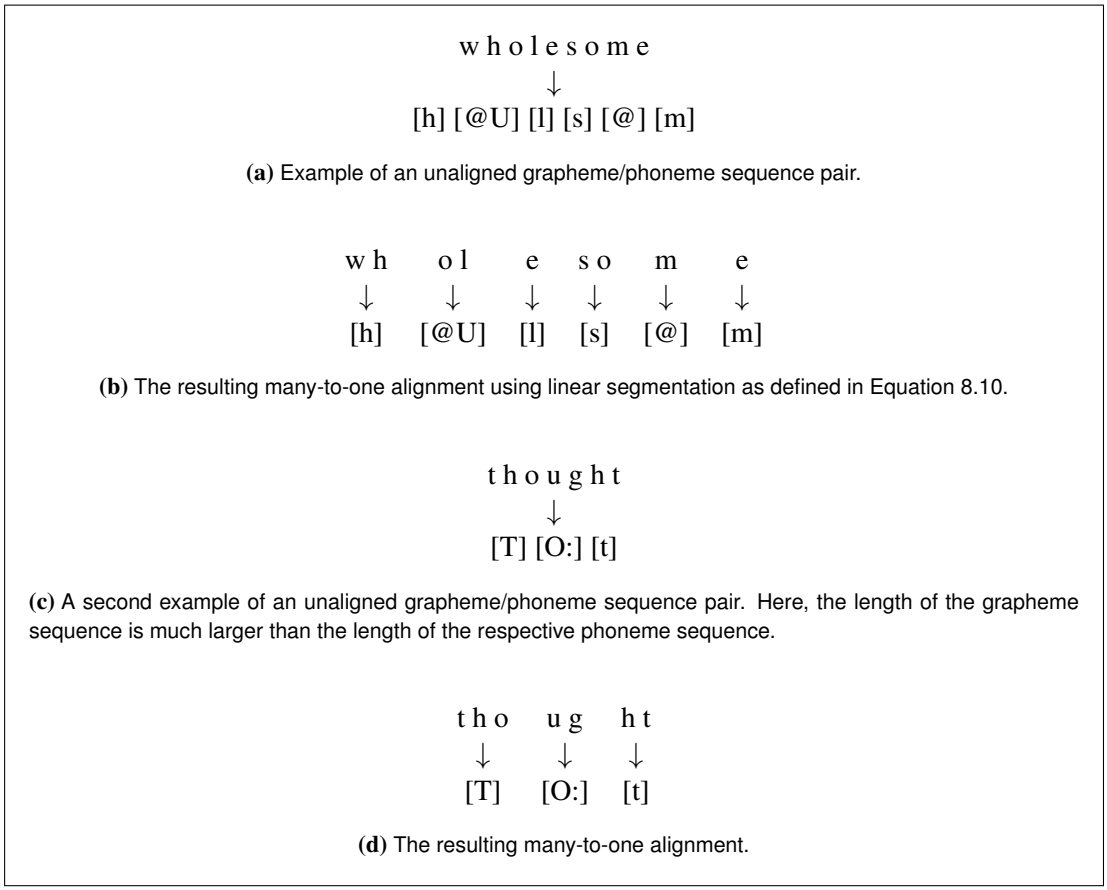
The linear segmentation applied is obtained by the following mapping from graphemes to phonemes, where each grapheme position is assigned to a phoneme position. For a given absolute position of a grapheme g within the reference grapheme sequence $\mathbf{g}$, denoted as $\text{pos}(\mathbf{g})$, the corresponding phoneme position and thus the corresponding phoneme is returned. len() is a function returning the length of a given phoneme or grapheme sequence.

$$\text{pos}(\mathbf{g}) \rightarrow \left(\text{pos}(\mathbf{g}) \cdot \frac{\text{len}(\boldsymbol{\varphi})}{\text{len}(\mathbf{g})} \right) \mod \text{len}(\boldsymbol{\varphi}) \tag{8.10}$$

Some Examples of such a linear segmentation are given in Figure 8.2. In cases where the grapheme sequence is shorter than the phoneme sequence, i.e. $\text{len}(\mathbf{g}) < \text{len}(\boldsymbol{\varphi})$, filler graphemes are inserted. An example is given in Figure 8.3. Naturally, this process can only be done in the training data since the phoneme sequence has to be known beforehand.

8.2.2 Alignment from giza++

Since an alignment between source and target side is also generated within the SMT process, we used this well-known framework to generate alignments for the G2P task. giza++ is a popular tool to derive alignments from unaligned data using the IBM models 1, 3 till 5 [Brown & Della Pietra$^+$ 93] as well as the HMM model as described in [Vogel & Ney$^+$ 96]; cf. also Section 1.1.1 for more details about the SMT process. The approach is as follows: on the (unaligned) training data, IBM 1 is applied for 4 iterations. The HMM model is initialized with the resulting alignment and training continues for another 4 iterations. With the HMM alignment, the IBM 3 model is initialized and trained for another 2 iterations. IBM 4 and IBM 5 follow in the same manner for 2 and 3 iterations respectively. Since the used alignment models produce only one-to-many alignments, IBM1 till 5 and the HMM model are also trained in the inverse direction, i.e. switching target and source side. Usually, the alignments in both directions are interpolated using some heuristics to lead to the final alignment. We applied the



(a) Example of an unaligned grapheme/phoneme sequence pair.

(b) The resulting many-to-one alignment using linear segmentation as defined in Equation 8.10.

(c) A second example of an unaligned grapheme/phoneme sequence pair. Here, the length of the grapheme sequence is much larger than the length of the respective phoneme sequence.

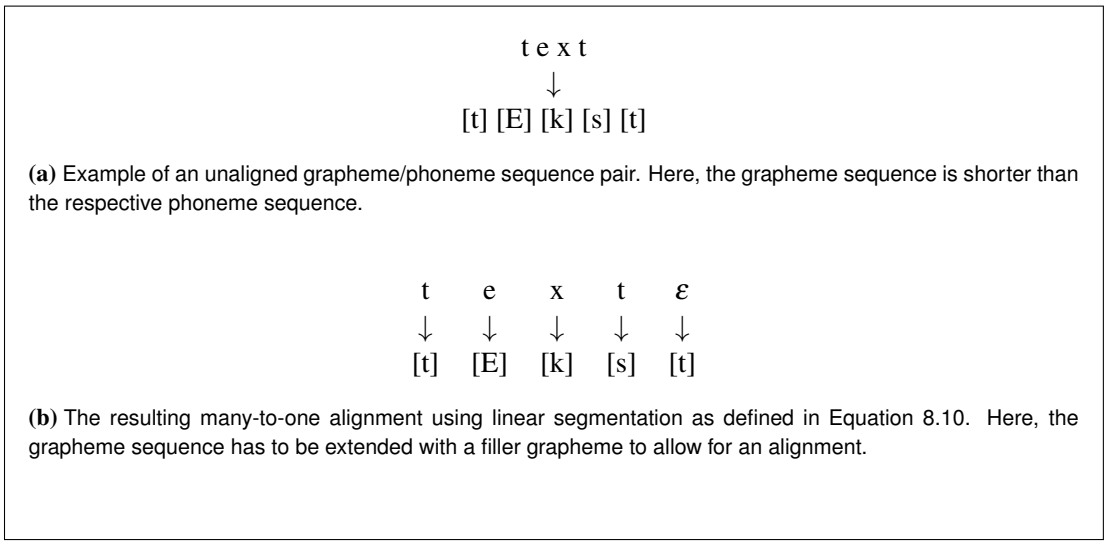
(d) The resulting many-to-one alignment.

Figure 8.2 Example for a linear segmentation of a grapheme/phoneme sequence pair to get a many-to-one alignment for CRF training as described in Section 8.2.1. The necessary one-to-one alignment can be easily obtained by applying the BIO scheme.

method as describe in [Matusov & Zens+ 04]. Here, symmetric word alignments are calculated which allow for alignments where each source word and each target word is aligned at least once.

Instead of directly using the alignments from the IBM models, we used the resulting γ or cost matrices which contain the weights resp. probabilities for the various alignment points of the training corpus. We combined these gammas from the IBM 5 model with the gammas from the IBM 4 model in both directions, i.e. source-to-target and target-to-source. Whereas the IBM 4 gammas are weighted by 0.5, the IBM-5 gammas are weighted by 1. Using these four knowledge sources, some heuristics are used which force the resulting alignment to be monotonous. Additionally, the insertion of an empty word is permitted. In Figure 8.4, examples for alignments using this giza++-based approach are presented.

As can be seen in the example illustrated in Figure 8.4 d, we are using statistical models which are error-prone. In this case, already the alignment generated by giza++ has been erro-



(a) Example of an unaligned grapheme/phoneme sequence pair. Here, the grapheme sequence is shorter than the respective phoneme sequence.

(b) The resulting many-to-one alignment using linear segmentation as defined in Equation 8.10. Here, the grapheme sequence has to be extended with a filler grapheme to allow for an alignment.

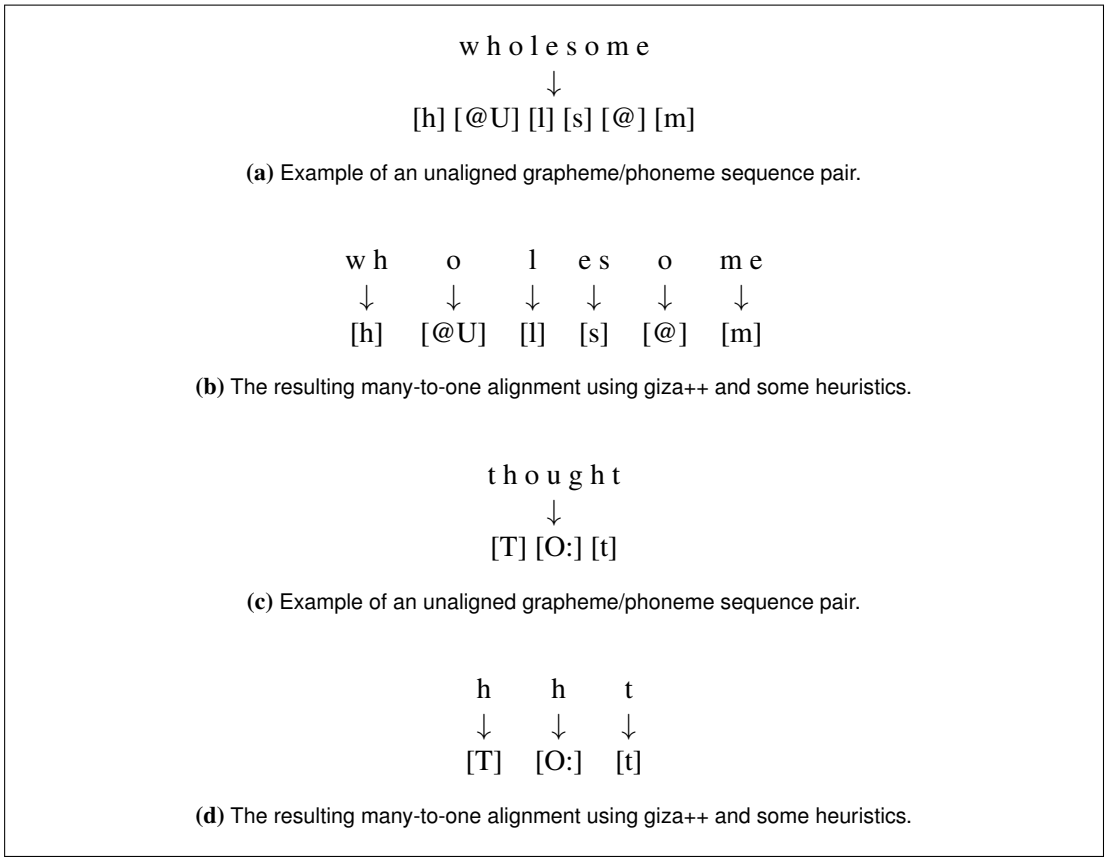
Figure 8.3 Example for a linear segmentation of a grapheme/phoneme sequence pair to get a many-to-one alignment for CRF training as described in Section 8.2.1. Here, a filler grapheme has to be inserted to obtain a valid alignment.

neous, most likely since the lengths of target and source side differ greatly. But in most cases, the alignment quality is better than using linear segmentation, as can be seen e.g. in Figure 8.4 b compared to Figure 8.2 b.

8.2.3 Alignment from Joint-Multigram Approach

Within the joint-*n*-gram approach as presented in Section 6.1.4, an alignment is implicitly trained. It is possible to use the trained model to retrieve the (single-best) alignment. The alignment reflects the grapheme/phoneme length constraints within a graphone, which have been introduced as parameter L within the aforementioned Section 6.1.4. Usually, for Western-European languages, best results are obtained when the graphone length is restricted to zero to one on both, grapheme and phoneme side. Thus, the insertion of empty graphemes and phonemes is allowed, which might be helpful since length differences between source and target side could be compensated. Another helpful side effect: the resulting alignment is already a one-to-one alignment and can directly be used for CRF training without introducing phoneme tags using the BIO scheme. Examples for such alignments are presented in Figure 8.5.

Concerning the grapheme side, we found it useful to introduce so-called named-epsilons whenever the G2P model hypothesizes an empty grapheme. Instead of only one (global) empty grapheme token, the empty graphemes are "named" with their predecessor grapheme and thus clustered. This approach corresponds to the BIO scheme, but on grapheme side. The disadvantage of this method is that the input vocabulary size grows by a factor of up to two, although this is not critically for CRF training. In Figure 8.6 b, a named-epsilon is used to compensate for a grapheme which maps to two phonemes. In Figure 8.6 d, an effect is illustrated which



(a) Example of an unaligned grapheme/phoneme sequence pair.

(b) The resulting many-to-one alignment using giza++ and some heuristics.

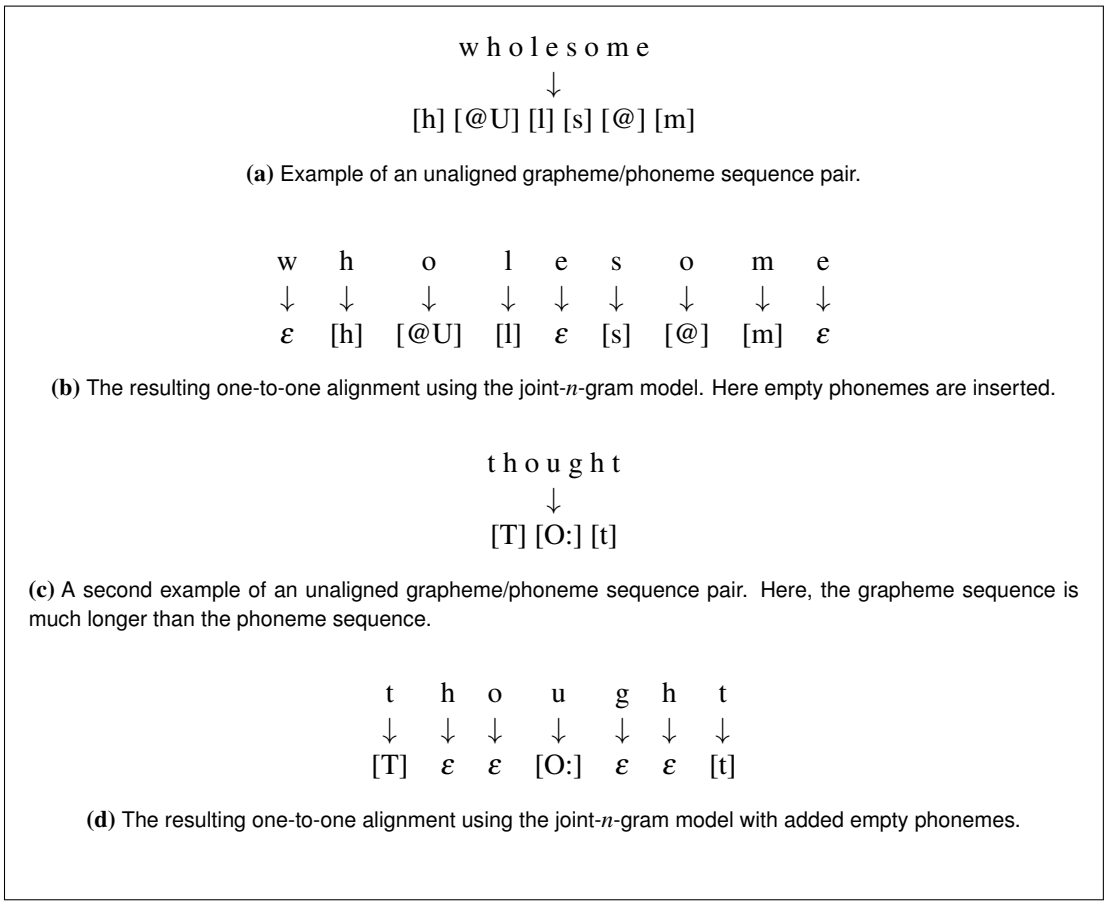
(c) Example of an unaligned grapheme/phoneme sequence pair.

(d) The resulting many-to-one alignment using giza++ and some heuristics.

Figure 8.4 Example for a giza++-based alignment of a grapheme/phoneme sequence pair to get a many-to-many alignment for CRF training as described in Section 8.2.2. The necessary one-to-one alignment can be easily obtained by applying the BIO scheme.

might seem strange at first. On both, grapheme and phoneme side, epsilons are inserted which in some way cancel each other out, i.e. the grapheme/phoneme sequence is enlarged by one symbol. This can happen due to the statistical nature of the approach and if similar occurrences have been observed in the training data where there is an advantage in modeling this event in exactly this way.

Although the performance of the model is quite good, there is another disadvantage in introducing epsilons on source side. If we want to apply a trained CRF model on a grapheme sequence, we have first to apply the original joint-*n*-gram model to this sequence since we have to retrieve the grapheme sequence with possible named-epsilons. If we do not include epsilons on source side, results are considerably worse as has been shown in [Guta 11].



(a) Example of an unaligned grapheme/phoneme sequence pair.

(b) The resulting one-to-one alignment using the joint-n-gram model. Here empty phonemes are inserted.

(c) A second example of an unaligned grapheme/phoneme sequence pair. Here, the grapheme sequence is much longer than the phoneme sequence.

(d) The resulting one-to-one alignment using the joint-n-gram model with added empty phonemes.

Figure 8.5 Examples for alignments based on the joint-n-gram model. Due to the parameter settings, one-to-one alignments are generated directly.

8.3 EM-Style Alignment (Maximum Approach)

Since the goal is to integrate the alignment within CRF training, a first step would be to do without an external model to pre-calculate the alignment but use CRFs instead to iteratively improve an initial alignment which has been generated by linear segmentation as proposed in Section 8.2.1. This obviously does not include any additional knowledge sources and the idea is very similar to the linear segmentation applied in ASR needed to bootstrap the monophone models. There, the initial alignment between transcription and audio signal is generated by equally distributing the phonemes over the duration of the audio. Using this initial segmentation, we first train a CRF model with a reduced feature set (i.e. only the lexical feature at the current position and the bigram feature) for a limited number of iterations. We then apply the resulting reduced CRF model to the training data which results in a new and hopefully improved alignment and start the model training again from scratch, i.e. we reset the feature weights Λ to

e x e r c i s e s
↓
[E] [k] [s] [@] [s] [aI] [z] [I] [z]

(a) Example of an unaligned grapheme/phoneme sequence pair.

e	x	ε_x	e	r	c	i	s	ε_s	e	s
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
[E]	[k]	[s]	[@]	[ε]	[s]	[aI]	[ε]	[z]	[I]	[z]

(b) The resulting one-to-one alignment using the joint-n-gram model. Here, named-epsilons on grapheme side are introduced to compensate for a grapheme which aligns to two phonemes.

y e s
↓
[j] [E] [s]

(c) A second example of an unaligned grapheme/phoneme sequence pair.

y	e	s	ε_s
↓	↓	↓	↓
[j]	[E]	[ε]	[s]

(d) The resulting one-to-one alignment using the joint-n-gram model. Here, an empty grapheme and an empty phoneme have been inserted in such a way that both, the grapheme and phoneme sequence grow by one symbol.

Figure 8.6 Examples for alignments based on the joint-n-gram model. Due to the parameter settings, one-to-one alignments are generated. Some strange phenomena are shown in Figure b and Figure d, where filler graphemes and phonemes have been added.

zero and we also reset the step size for the RProp optimization.

This procedure is possible since CRFs need and generate one-to-one alignments which can be easily exchanged for one another. It is necessary to use a reduced feature set since CRFs converge very fast and using complex feature sets would lead to sharp models which would only reproduce the alignment present in the training data. Since we need a certain amount of flexibility and alternative hypotheses in the search space, we need broad models. This is also the reason for the very limited number of training iterations. A comparison can be drawn to discriminative training in ASR. There, the search space for the competing hypotheses is generated by word lattices which are built using a broad unigram LM instead of a much sharper tri- or fourgram LM [Schlüter & Müller$^+$ 99]. This process of iteratively improving the alignment has some similarities to the EM algorithm, whereas the maximization or M-step is the training of the reduced model and the expectation or E-step corresponds to resegmenting the training

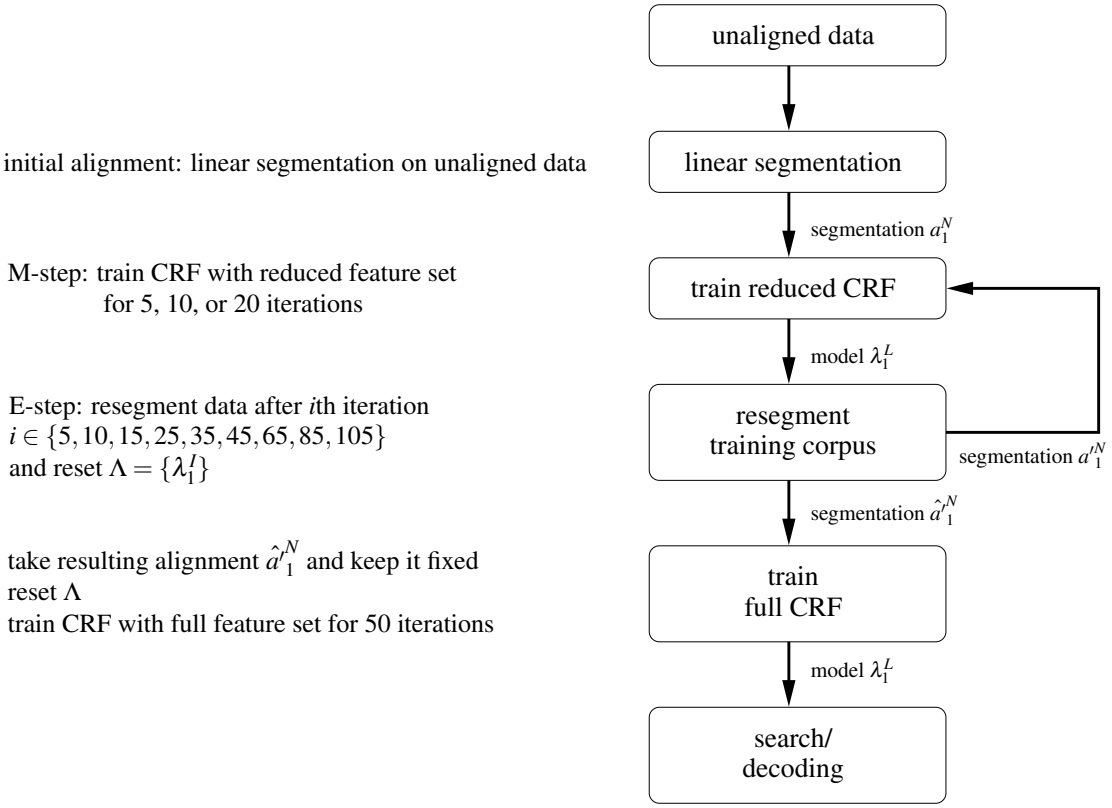


Figure 8.7 Flow chart for the EM-style alignment (maximum approach).

corpus after some iterations. We empirically derived and used the following setting: in total, we train the reduced CRF alignment model for 105 iterations, whereas we perform a resegmentation and resetting of feature weights after iterations 5, 10, 15, 20, 25, 35, 45, 65, 85, and 105. The resulting alignment is kept fixed and used to train a CRF model with a full feature set from scratch for 50 iterations. A flow chart of the resulting algorithm is given in Figure 8.7.

In Figure 8.8, there are some examples showing iteratively improving alignments.

Concerning the convexity of this approach, an overall global optimum can not be guaranteed anymore. For the reduced CRF model, the optimization is convex until the resegmentation step, which breaks convexity. Thus, the alignment process is only locally convex. For the full CRF model training based on the fixed alignment, the optimization is again globally convex.

8.4 Alignment as Hidden Variable within CRFs: Restricted HCRFs (Summation Approach)

Although there is no additional or external model needed anymore to pre-compute the alignment, the final alignment is still precomputed and iteratively improved within the approach presented in the previous section. A theoretically sound integration of the alignment would be

	t	h	o	u	g	h	t
	↓	↓	↓	↓	↓	↓	↓
iteration 0	T	T	T	O:	O:	t	t
iteration 5	T	T	O:	O:	O:	t	t
iteration 15	T	T	O:	O:	O:	O:	t

(a) Example for an iteratively improved alignment. After iteration 15, there has been no change in the alignment due to resegmentation.

	w	h	o	l	e	s	o	m	e	n	e	s	s
	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
iteration 0	h	@U	@U	@U	l	l	s	@	m	m	n	I	s
iteration 5	h	h	@U	l	l	s	@	@	m	n	n	I	s
iteration 10	h	h	@U	l	s	s	@	m	m	n	I	s	s
iteration 45	h	h	@U	l	s	s	@	m	n	n	I	s	s

(b) Another example of an iteratively improved alignment. Here, it took 45 iterations until convergence.

Figure 8.8 Examples for alignments based upon CRFs bootstrapped with linear segmentation and following the EM maximum approach. For clarity, the start/continue tags as well as the square brackets around the phonemes have been omitted.

to train the model parameters $\Lambda = \{\lambda_1^M\}$ as given in Equation 8.11 without any pre-computation phase.

$$Pr(\varphi_1^N \mid g_1^L) = \sum_{a_1^L} Pr(\varphi_1^N, a_1^L \mid g_1^L) \tag{8.11}$$

$$\approx \sum_{a_1^L : t_1^L = (t_1^N, a_1^L)} p_{\Lambda,\mathrm{CRF}}(t_1^L | g_1^L) \tag{8.12}$$

$$= \frac{\sum_{t_1^L \in t(\varphi_1^N)} \prod_{l=1}^{L} \exp\left(\sum_{m=1}^{M} \lambda_m f_m(t_{l-1}, t_l, g_1^L) \right)}{\sum_{\tilde{t}_1^L} \prod_{l=1}^{L} \exp\left(\sum_{m=1}^{M} \lambda_m f_m(\tilde{t}_{n-1}, \tilde{t}_n, g_1^L) \right)} \tag{8.13}$$

One approximation would be to restrict the alignment to all possible tag sequences $t_1^L \in t(\varphi_1^N)$ leading to a many-to-one alignment between graphemes and phonemes (cf. Equations 8.12 and 8.13). This type of alignment is easily realized within our software since it has been designed to solve the tagging task, where only many-to-one alignments occur. The disadvantage is that it is still not possible to model other alignment types, e.g. many-to-many alignments. As for all presented methods so far, only many-to-one alignments would be possible and thus still not the

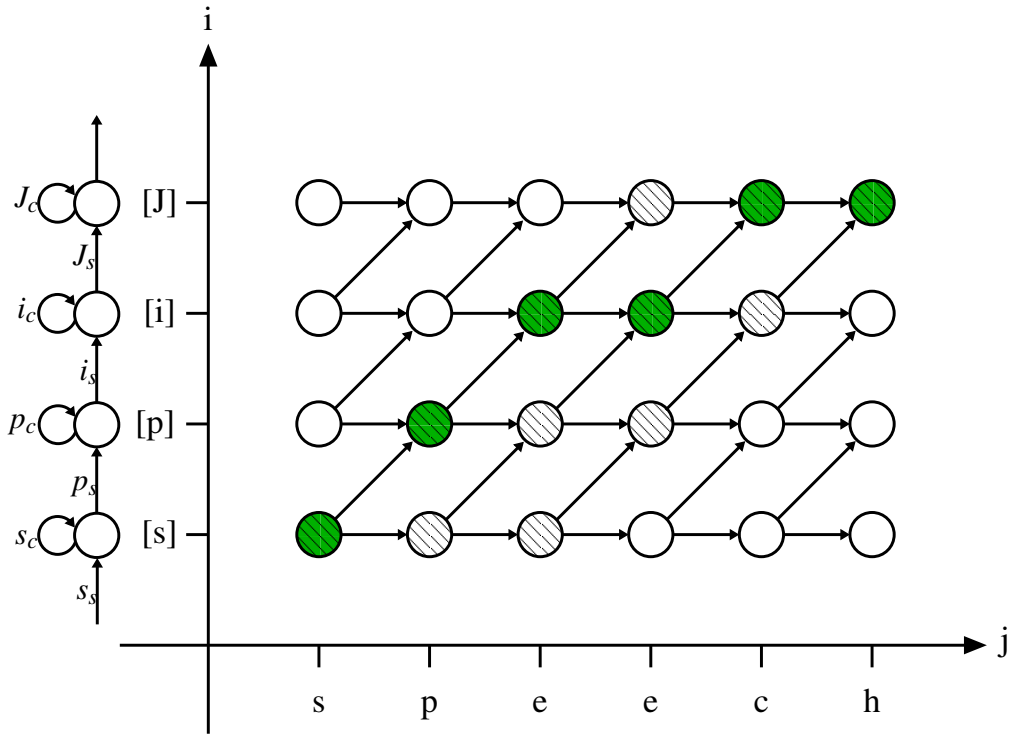


Figure 8.9 Developed view of a 0-1-HMM leading to a many-to-one alignment. A node in this graph at position (j,i) represents the alignment $a_j = i$ meaning that g_j is aligned to φ_i. All alignment points which are theoretically allowed by the constraints from Equations 8.1 - 8.3 are hatched while the correct alignment is additionally green/shaded.

mapping from e.g. the letter "x" to the two phonemes "[k]" and "[s]".

In Figure 8.9, theoretically possible alignments using phoneme tags are shown. Here, the HMM on the left represents the possible monotone alignment steps: either a grapheme g_j at position j is aligned to the currently "active" phoneme at position $\varphi_{a_{j-1}}$ (loop transition) or to the next phoneme at position $\varphi_{a_{j-1}+1}$(forward transition). Since only loop and forward transitions are allowed, this model is usually referred to as 0-1-HMM. The graph on the right depicts the developed view of this HMM. A node in this graph at position (j,i) represents the alignment $a_j = i$ meaning that g_j is aligned to φ_i.

Certainly, not all nodes in the graph in Figure 8.9 are necessarily part of a valid alignment. This highly depends on the size of the grapheme and phoneme sequence and also on the constraints for monotonic alignments as presented in Equations 8.1, 8.2 and 8.3. Within the figure, all theoretically valid alignment points w.r.t. the alignment constraints are hatched while the correct alignment is additionally green/shaded.

The automaton depicted in Figure 8.10 shows the topology of all possible tag sequences t_1^6 for the word/pronunciation pair "speech/[spiJ]". The correct path is marked with green/shaded states. The states are named with the last emitted phoneme symbol whereas the arcs are labeled with the currently read grapheme followed by a double point as separator followed by

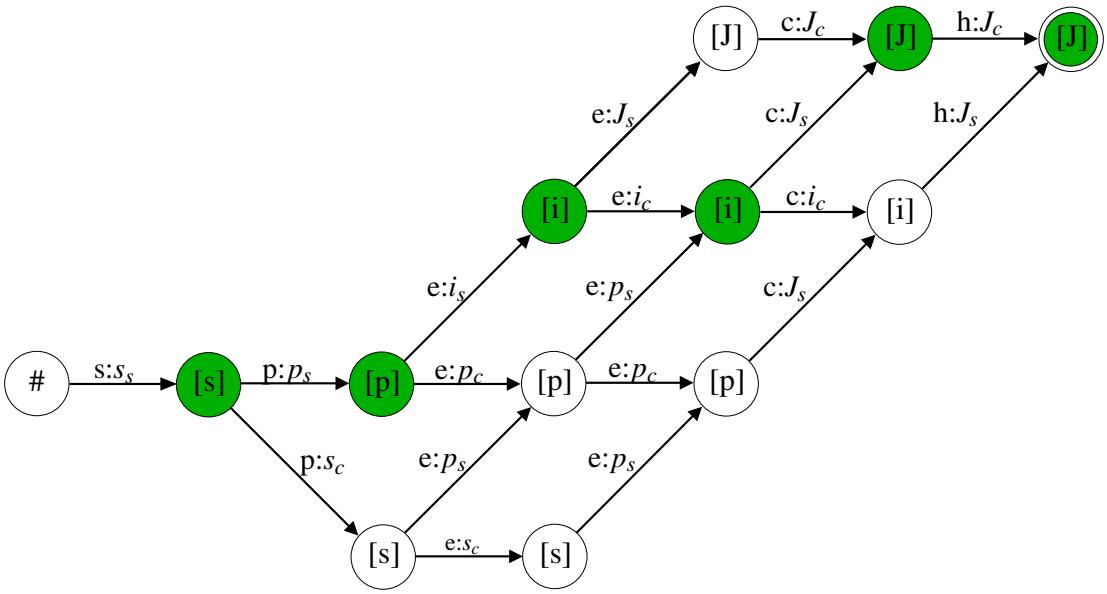


Figure 8.10 FSA showing the topology of all possible tag sequences t_1^6 for the word/pronunciation pair "speech/[spiJ]". The correct path is marked with green/shaded states. The states are named with the last emitted phoneme symbol whereas the final state is marked with a double circle. The first state is a virtual start state for modeling reasons. The transitions are labeled with a source symbol followed by a colon as separator and a phoneme tag. Here, the indices "s" and "c" represent the start and continue markers.

a phoneme tag. The final state is marked with a double circle. The first state marked with a hashtag is a virtual start state for modeling reasons. This automaton can be regarded as the developed view of the 0-1-HMM shown in Figure 8.9 respecting the alignment constraints for an actual grapheme/phoneme sequence pair.

Integrating the alignment as a hidden variable has a number on consequences for CRF training. For instance, we get a non-convex optimization problem which might get stuck in local optima. Also, training time will increase due to the additional sum in the numerator (cf. Equation 8.13). Since this sum is a restricted version of the sum in the denominator, the computational cost is doubled in worst case. Considering decoding, Bayes' decision rule is given in Equations 8.14.

$$\begin{aligned}\hat{\varphi}_1^N &= \underset{\varphi_1^N}{\operatorname{argmax}} \left\{ Pr(\varphi_1^N | g_1^L) \right\} \\ &= \underset{\varphi_1^N}{\operatorname{argmax}} \left\{ \sum_{a_1^L} Pr(\varphi_1^N, a_1^L | g_1^L) \right\} \\ &= \underset{\varphi_1^N}{\operatorname{argmax}} \left\{ \sum_{a_1^L : t_1^L = (\varphi_1^N, a_1^L)} Pr(t_1^L | g_1^L) \right\}\end{aligned} \tag{8.14}$$

As can be seen, there is a mixture of summation and maximization which results in an exponential complexity. Thus, it is hard to use the sum in search, as depicted in Equation 8.15.

$$\begin{aligned}\hat{\varphi}_1^N &= \underset{\varphi_1^N}{\operatorname{argmax}} \left\{ \sum_{a_1^L : t_1^L = (\varphi_1^N, a_1^L)} p_{\Lambda,CRF}(t_1^L | g_1^L) \right\} \\ &= \underset{\varphi_1^N}{\operatorname{argmax}} \left\{ \sum_{a_1^L : t_1^L = (\varphi_1^N, a_1^L)} \prod_{l=1}^{L} \exp\left(\sum_{m=1}^{M} \lambda_m f_m(t_{l-1}, t_l, g_1^L) \right) \right\}\end{aligned} \tag{8.15}$$

We apply maximization for the alignment in search, besides the Viterbi decoding, which is also a replacement of the sum over all phoneme sequences with a maximization. The final decoding is given in Equation 8.16.

$$\begin{aligned}\hat{\varphi_1^N} &= \underset{\varphi_1^N, a_1^L}{\operatorname{argmax}} \left\{ p_{\Lambda,CRF}(t_1^L | g_1^L) \right\} \\ &= \underset{t_1^L : t_1^L = (\varphi_1^N, a_1^L)}{\operatorname{argmax}} \left\{ \prod_{l=1}^{L} \exp\left(\sum_{m=1}^{M} \lambda_m f_m(t_{l-1}, t_l, g_1^L) \right) \right\}\end{aligned} \tag{8.16}$$

8.5 Experimental Comparison

We evaluated the presented alignment approaches on the NETtalk and CELEX databases. The respective corpora are presented in Sections A.2 and A.3. As baseline system for NETtalk, the same features as for the experiments described in Section 7.2.2 have been used: lexical features in a window of $[-4,\ldots,4]$ around the current word, the bigram features and source n-gram features of length two up to six. Additionally, the margin extension to the training criterion is used. We do not apply the EN, since the total number of features and training samples are comparatively small. Thus, we just have the L_2 regularization which has been set to 2^{-3}. A feature-build-up for the NETtalk corpus with a given manual alignment is presented in Table 8.1.

For the CELEX system, the feature setup is a little different. Since we already have the system optimized for NETtalk and we do not expect the optimal CELEX system too far away, we first used the optimized NETtalk features to tune the L_2 regularization on CELEX leading to 2^{-7}. Afterwards, we tuned the lexical feature window together with bigram feature. Compared to NETtalk, a bigger window of $[-5,\ldots,5]$ around the current word lead to best results. Finally, we tuned the source n-gram features. Here, using all n-grams of length two up to five resulted in the best performance. The feature build-up is summed up in Table 8.2.

As usual PER as well as WER are used to measure the performance of the various systems. Using these system setups, the various alignment methods have been tested. The results are given in Table 8.3.

Concerning the various external models, the joint-n-gram alignment clearly outperforms the other two methods when used to train a CRF model. It was predictable that the linear alignment

Table 8.1 Feature build-up on the NetTalk 15k corpus: Experimental results for various features and their combinations. Here, a manual alignment has been used.

features	PER [%]		WER [%]	
	Dev	Eva	Dev	Eva
source lexicals	14.6	14.6	59.7	57.5
+ unigram	14.8	14.6	60.2	57.6
+ source n-grams	8.0	8.3	35.0	36.0
+ bigram	12.3	12.2	51.9	49.5
+ source n-grams	**7.4**	**7.9**	**33.2**	**34.2**

Table 8.2 Feature build-up on the Celex corpus: Experimental results for various features and their combinations. Here, an alignment provided by the joint-n-gram approach has been used.

features	PER [%]		WER [%]	
	Dev	Eva	Dev	Eva
source lexicals + unigram	12.1	12.2	55.9	56.7
source lexicals + bigram	9.2	10.4	44.8	48.9
+ source n-grams	**2.6**	**2.5**	**13.0**	**12.4**

Table 8.3 Experimental results for various alignments for CRF training on NetTalk 15k and Celex. "CRF max" represents the EM-style maximum approach whereas CRF sum stands for the respective summation approach. For comparison, the results using the Sequitur tool from [Bisani & Ney 08] are also reported.

						[Bisani & Ney 08]	
data set	alignment	PER [%]		WER [%]		PER [%]	WER [%]
		Dev	Eva	Dev	Eva	Eva	Eva
NETtalk 15k	linear	10.1	10.6	43.7	44.9		
	giza++	7.6	8.0	33.9	34.5		
	joint n-gram	7.4	7.9	33.2	34.2	8.3	33.7
	manual	7.6	7.8	33.6	33.7		
	CRF max	7.5	7.9	34.0	34.1		
	CRF sum	9.1	9.6	40.7	40.2		
CELEX	linear	5.3	4.9	25.1	23.6		
	giza++	3.7	3.6	18.8	18.1		
	joint n-gram	2.6	2.5	13.0	12.4	2.5	11.4
	CRF max	2.9	2.8	14.6	13.9		
	CRF sum	3.7	3.5	17.8	16.8		

would lead to poor results due to its very limited modeling power. With respect to the translation model, the performance on the NETtalk corpus is quite close to the performance of the joint-n-gram model. One reason for this is that the corpus itself is designed in a way that mostly one-to-one alignments and especially many-to-one alignments suffice. Since the translation models underlying giza++ fit into this framework, the good performance can be explained which leads to results which are quite as good as with using the manual alignment. Due to the possibility of the joint-n-gram model to insert epsilons on grapheme and phoneme side, the best results can be achieved. This comes at the prize of having to train and run two models on the same data. Compared to the results reported in [Bisani & Ney 08], the CRF results based on giza++ and joint-n-gram alignment outperform the reported results by roughly 3–4% relative. This can be explained with the rather small size of the training set and the fact that discriminative models usually outperform generative ones in such a setting (cp. Section 7.2.2).

On the CELEX corpus, the ranking of the approaches is the same. The linear segmentation leads to poor results. Here, the gap between giza++ and joint-n-gram alignment is greater than for the NETtalk corpus. One reason for this is the fact that it is easier for the latter approach to model many-to-many alignments as for giza++ and this type of alignment is needed to reflect the correct relationship between graphemes and phonemes. Although the joint-n-gram alignment is again the best one, it does not outperform the results from the Sequitur model, which is slightly better, especially when considering the word error rate. One reason might be that the CELEX training corpus is much larger and thus the difference in performance of discriminative versus generative training criteria is not this big.

If we take a look at the integrated alignment approaches based on the EM algorithm and the summation approach, i.e. CRF max and CRF sum in the table, they do not outperform a given alignment from an external model. On NETtalk, the maximum approach leads to results similar to the results of the joint-n-gram and the manual alignment. Thus, it is possible to bootstrap a model with a linear alignment, which lead to 10.6% PER on the evaluation set, and iteratively improve the alignment and the model by 25% relative down to 7.9% PER. Concerning the summation approach, the results are roughly 20% relative worse than for the joint-n-gram approach but still better than the linear segmentation. One reason behind the poor performance might be the non convex training criterion which might have got stuck in a local optimum. On CELEX the performance of the maximum approach is close to the best performing joint-n-gram alignment (roughly 10% relative deterioration). The summation approach is again worse but this time close to (or marginally better than) the giza++ model. Overall, the maximum approach seems to be the better choice if an alignment integrated into CRF is desired.

8.6 Alignment as Hidden Variable within CRFs: HCRFs

It is possible to overcome the disadvantages of using the restricted HCRF as presented in Section 8.4. At least for the G2P task it suffices to have one-to-two alignments, e.g. to align the letter “x” to the phoneme sequence “[k][s]” which occurs frequently within the CELEX database. Allowing for those alignment links, it is possible to correctly align a grapheme sequence g_1^L to a phoneme sequences φ_1^N where the phoneme sequence is longer than the grapheme sequence,

i.e. $N > L$. Thus, we would be able to model many-to-many alignments and would not need the restriction from the previous section. The general formula is given in Equations 8.17 - 8.19.

$$Pr(\varphi_1^N \mid g_1^L) = \sum_{a_1^L} Pr(\varphi_1^N, a_1^L \mid g_1^L) \tag{8.17}$$

$$= \sum_{a_1^L} p_{\Lambda,\mathrm{CRF}}(\varphi_1^N, a_1^L \mid g_1^L) \tag{8.18}$$

$$= \frac{\sum_{a_1^L} \prod_{n=1}^{N} \exp\left(\sum_{m=1}^{M} \lambda_m \cdot h_m(a_l, \varphi_{n-1}, \varphi_n, g_1^L)\right)}{\sum_{\tilde{a}_1^L} \sum_{\tilde{\varphi}_1^N} \prod_{n=1}^{N} \exp\left(\sum_{m=1}^{M} \lambda_m \cdot h_m(\tilde{a}_l, \tilde{\varphi}_{n-1}, \tilde{\varphi}_n, g_1^L)\right)} \tag{8.19}$$

Approaches dealing with HCRFs have been proposed in the literature. In e.g. [Koo & Collins 05], a POS parse tree is used to represent the graph over which the sum is accumulated. The resulting model is used for reranking. The authors of [Quattoni & Wang+ 07] use a mesh between features as graph structure. Similar to the presented approach, but only applied to POS tagging and named entity recognition (e.g. with a small output vocabulary), is the work described in [Yu & Lam 08]. The alignment is introduced as a hidden variable and in training the sum over all alignments is computed. For efficiency reasons, the number of hidden alignment states per state is restricted.

In ASR, usually 0-1-2 HMMs are used to model the relation between phonemes and the feature vectors (cf. Section 1.3.3). Whereas loop and forward transitions are used in the same sense then for the summation approach presented in Section 8.4, the skip transitions are used to compensate fast speaking rates and do thus skip phonemes. For G2P, we still do want to have monotone alignments and do not wish to skip phonemes. But we can use the skip arc mechanism in our favor. Instead of skipping a phoneme, we will produce two phonemes on one arc while consuming only one grapheme. This can be interpreted as introducing named-epsilons on grapheme side. An example for such an alignment is given in Figure 8.11.

Basically, the automaton presented in Figure 8.9 has been extended by the proposed skip arcs. Now it is possible to e.g. align the letter “x” to the two phonemes “[k][s]”. Within the developed view, the correct alignment path is marked in green/shaded. The implicitly aligned phoneme “k” is marked in yellow/shaded differently.

As for the 0-1 HMM approach presented in the previous chapter, the training criterion is again not convex and thus a global optimum not guaranteed. It is possible to improve the stability of a HCRF by initializing the lexical (φ_n, g_{a_n}) features with IBM-scores, as presented e.g. in [Guta 11]. Here, the IBM-1 scores are calculated in a preprocessing step on the training data and are then directly used as feature weights. For all experiments reported using the 0-1-2 HMM, IBM-1 initialization is applied. Additionally and similar to ASR, it is necessary to introduce transition penalties $\delta_0, \delta_1, \delta_2$ for the loop, forward and skip transitions. These penalties can be either chosen empirically (tuning on the development set) or they could be integrated as features within the CRF framework. We have chosen the latter approach since it seems to lead to better results (cf. [Lehnen & Hahn+ 12]). More details about this method, especially with respect to implementation, are presented in [Lehnen & Hahn+ 11a], [Lehnen & Hahn+ 12] and [Guta 11].

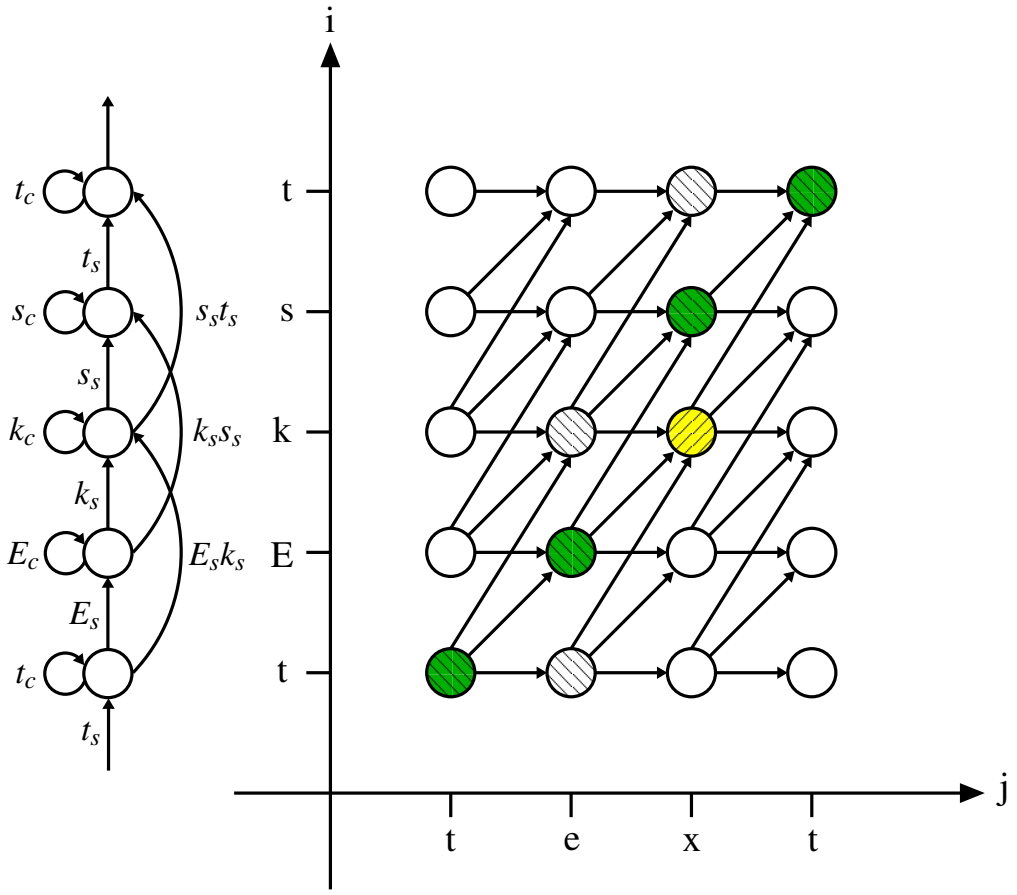


Figure 8.11 Developed view of a 0-1-2-HMM leading to a many-to-many alignment. A node in this graph at position (j,i) represents the alignment $a_j = i$ meaning that g_j is aligned to φ_i. All alignment points which are theoretically allowed by the constraints from Equations 8.1 - 8.3 are hatched while the correct alignment is additionally green/shaded. Since the skip-arcs produce two phoneme tags, the respective second alignment point which is implicitly generated is yellow/shaded differently.

8.6.1 Experimental Results

Since the effect of the introduced skip arcs will only have an impact on data where many-to-many alignments are needed to model the correct relation between graphemes and phonemes, we only performed experiments on the CELEX corpus. A feature-build up is reported in Table 8.4.

We used the same feature set as used within the previous set of experiments, i.e. lexical features is a window of $[-5,\ldots,5]$ around the current word, the bigram feature and source n-gram features utilizing all n-grams of length two up to five. Additionally, a prior feature is used to weight the various transition types (loop, forward, skip). Since the overall system has gotten more complex and the automatic tuning of the HMM weights/features does not seem to converge as fast as for the other features, we used 75 iterations for the experiments. Due to the

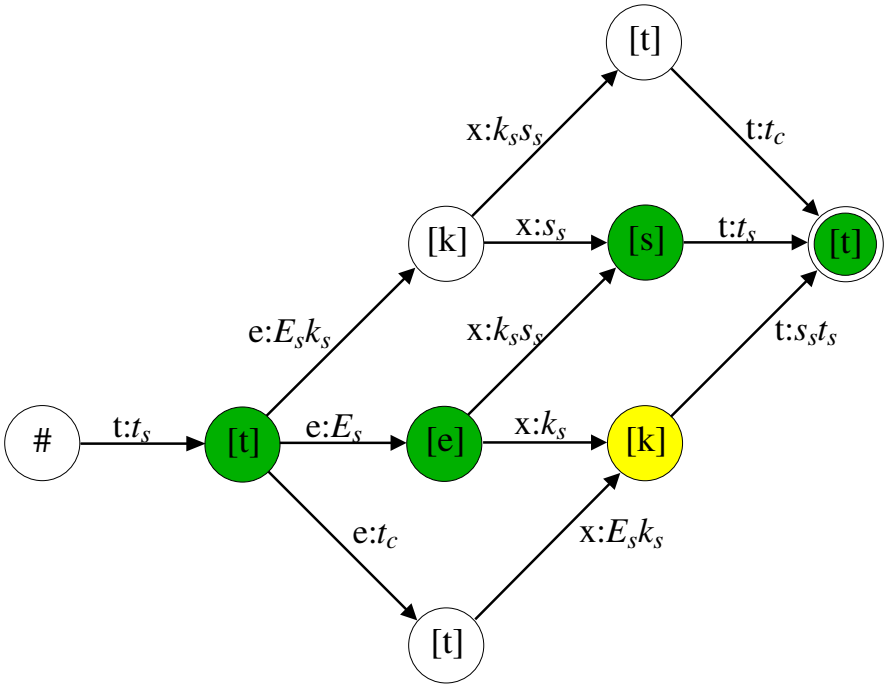


Figure 8.12 FSA showing the topology of all possible alignment paths for the word/pronunciation pair "text/[tEkst]". The correct path is marked with green/shaded states. The implicitly produced "k" state via the skip arc is marked yellow/differently shaded. The states are named with the last emitted phoneme symbol whereas the final state is marked with a double circle. The first state is a virtual start state for modeling reasons.

Table 8.4 Feature build-up for the 0-1-2 HMM alignment (HCRF) on the Celex corpus.

	PER[%]		WER[%]		number of	total number
	Dev	Eva	Dev	Eva	active features	of features
(g_{a_n}, φ_n) + prior	52.5	52.7	97.1	97.7	1,265	1,566
+ source n-grams	4.0	3.8	20.9	20.2	9,603,635	71,794,024
+ $(\varphi_n, \varphi_{n-1})$	2.6	2.5	12.6	12.3	9,605,051	71,797,040

high number of feature functions, we applied a simple feature filter, where we only considered all features seen at least once in the training data additionally to EN. We empirically optimized the regularization parameters leading to $L_1 = L_2 = 2^{-4}$. Besides the total number of features, the number of features with a non-zero feature weight ("active") is additionally given in the table. Compared to the summation approach presented in Table 8.3, there is a big improvement with respect to performance. Now, the necessary alignment links can be modeled correctly and the error rate improves from 3.5% PER on the evaluation set down to 2.5% PER. Due to the various feature filter techniques, the number of active features for the best performing system comprises only roughly 13% of the total number of features. The PER does now favorably compare with the results presented in [Bisani & Ney 08], while the WER is still a little bit worse, meaning that the errors from the CRF system are spread across more words than for the joint-n-gram approach.

8.7 Conclusion

In this chapter, we have discussed several possibilities to train CRFs when no alignment between graphemes and phonemes is given with the training data. For the G2P task, constraints can be put onto the alignment process such that no general many-to-many alignment but a monotonic alignment obeying linguistic characteristics suffices. In a first set of experiments, external models have been used to generate and provide a many-to-one respectively one-to-one alignment for CRF training. The alignment generated by the joint-n-gram model have achieved the best results on the NETtalk and the CELEX database. Using an EM like procedure, it is possible to use CRFs to generate an alignment using a linear segmentation of the training data as bootstrap. Despite the more complex and time consuming training process, this methods leads to comparatively good results. With a simple 0-1 HMM, it is possible to directly generate an alignment implicitly within CRF training (restricted HCRF) without any initialization procedure, but the results are significantly worse than for the externally provided alignment, especially on the CELEX corpus where one-to-two alignment links are needed to correctly model the dependencies between graphemes and phonemes. This can be achieved by extending the 0-1 HMM to a 0-1-2 HMM, where the conventional skip arcs are used to model the production of two phonemes while consuming only one grapheme. Using this method, it was possible to achieve the same performance as the joint-n-gram approach as presented in [Bisani & Ney 08] without an alignment provided by an external tool. To allow for a reasonable training time, additional features for the HMM penalties as well as feature count cut-offs have been introduced. To improve model accuracy, the lexical (φ_n, g_{a_n}) features are initialized using IBM-1 scores as initial feature weights.

8.8 Digression: Joint n-gram Features

In [Jiampojamarn & Cherry$^+$ 10], it has been reported that the so-called joint-n-gram features are helpful for the G2P task. They are basically defined as combining a source-n-gram feature

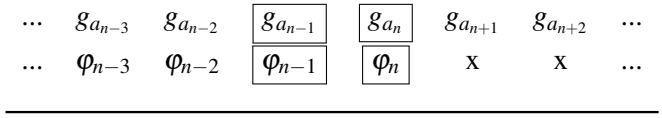


Figure 8.13 Example for a joint-*n*-gram feature.

and a target-*n*-gram feature to a joint *n*-gram, and thus conceptually similar to graphones. An example for such a feature is given in Figure 8.13.

Although the computational complexity of CRF models does not permit the use of longer contexts on phoneme side, it might be useful to build new features by combining a bigram feature and a source-*n*-gram feature. In the aforementioned figure, a combination of a bigram on grapheme side and a bigram on phoneme side is shown. We realized the joint-*n*-gram features in the following way: since there is only a bigram possible on phoneme side, this bigram is always part of a joint-*n*-gram. Additionally, the current position on source side is also always included. These three positions define the joint-*n*-gram with window size 0. With growing window size, each possible graphemic *n*-gram within the respective window forms a joint-*n*-gram feature, independently of its size and is included in the set of features. Since this procedure results in a very high number of features, additional feature selection mechanisms have to be applied. We tested various feature cut-off strategies and found that selecting a joint-*n*-gram feature if it does occur two times in the training data leads to good results. Additionally, all other features which are not joint-*n*-grams have to be observed at least once to be included.

8.8.1 Experimental Results

We performed a couple of experiments on the CELEX database to get an idea of the performance of the joint-*n*-gram features. As a baseline system, we used the best performing system so far, which has been presented in Section 8.6 and in the last line of Table 8.4 respectively. First, we optimized the window length. The results are presented in Table 8.5.

By using a window size of three, the PER can be improved by approximately 5% relative with respect to the baseline system. Note that although there seems to be not much difference with respect to performance on the development set, if considering the number of grapheme errors, the highlighted system has the least. On the development set, 0.1% PER corresponds to 35 errors. As can also be seen, the number of active features grows rapidly with a growing window size. In Table 8.6, an overview for the complete feature build-up for the HCRF system with joint-*n*-grams including the number of features is given for reference.

As can be seen, the number of theoretically possible features using the optimized joint-*n*-gram feature setup is roughly 178 million, more than doubling the number of features compared to the system setup without joint-*n*-grams. This clearly indicates that techniques are necessary to preselect features. Using our filtering approach together with EN, the number of active features is roughly 15 million, which is a reduction to around 8% of all possible features.

Next, we want to compare our best HCRF G2P system, with previously published results re-

Table 8.5 Tuning of the window size of the joint-n-gram features on the Celex corpus. As baseline system, the 0-1-2 HMM alignment HCRF from Section 8.6 has been used. Additional to the performance measures, the number of active features is also given.

window size	PER[%]		WER[%]		number of active features
	Dev	Eva	Dev	Eva	
no joint n-grams	2.6	2.5	12.6	12.3	9,605,051
0	2.5	2.5	12.3	12.1	9,627,902
1	2.5	2.4	12.0	11.9	9,926,207
2	2.5	2.5	12.1	12.0	11,234,722
3	2.5	**2.4**	12.0	**11.7**	14,608,700
4	2.5	2.5	12.2	12.1	20,914,427
5	2.5	2.5	11.9	12.0	30,522,151

Table 8.6 Feature build-up for the best performing HCRF system on Celex. Additionally to the PER and WER results, the number of active features as well as the number of all possible features are given.

	PER[%]		WER[%]		number of active features	total number of features
	Dev	Eva	Dev	Eva		
(g_{a_n}, φ_n) + prior	52.5	52.7	97.1	97.7	1,265	1,566
+ source n-grams	4.0	3.8	20.9	20.2	9,603,635	71,794,024
+ $(\varphi_n, \varphi_{n-1})$	2.6	2.5	12.6	12.3	9,605,051	71,797,040
+ joint-n-grams	2.5	2.4	12.0	11.7	14,608,700	177,916,276

spectively re-runs using the software from the original publications for the exact same CELEX data split, i.e. the exact same splitting in training, development and evaluation sets. The comparison is given in Table 8.7.

Concerning the presented results, the figures from [Bisani & Ney 08] and [Jiampojamarn & Cherry$^+$ 10] have been taken directly from the literature, since the numbers have been generated on the exact same splitting of the data as used for our experiments. The results from all other reported papers are reproductions using the original software. The available parameters have been tuned to the best of the author's knowledge on the development set (cf. Sections 6.1 and 6.2.2). The results are sorted with respect to decreasing performance on the evaluation set. As can be seen, the results from the HCRF approach compares favorably with the results reported in the literature. The method outperforms all other generative approaches, although the improvement over the joint-n-gram model from [Bisani & Ney 08] is not statistically significant with respect to PER; only the online discriminative training as presented in [Jiampojamarn & Cherry$^+$ 10] seems to be better, although the authors do only report the WER on the evaluation set. Concerning the comparison with the quite recent phonetisaurus tool as presented in [Novak 11], there is a quite big gap in performance. An updated version of this tool as presented in [Novak & Dixon$^+$ 12] now provides rescoring using neural networks and is expected to outperform the

Table 8.7 Comparison of various methods/tools on the Celex database. The results from [Bisani & Ney 08], [Jiampojamarn & Cherry+ 10] and have been taken from the literature, whereas for the other methods the experiments have been re-run using the original software from the respective publications. The results are sorted with respect to decreasing PER on the evaluation set.

method	PER[%]		WER[%]	
	Dev	Eva	Dev	Eva
[Chen 03]	3.9	3.9	17.3	17.4
[Novak 11]	3.6	3.4	18.0	16.7
[Kneser 00]	3.4	3.4	15.8	15.8
[Vozila & Adams+ 03]	3.5	3.4	15.5	15.2
[Bisani & Ney 08]	2.7	2.5	11.8	11.4
HCRF	2.5	2.4	11.8	11.7
[Jiampojamarn & Cherry+ 10]				10.8
[Novak & Dixon+ 12]				

older version. There are no results reported on the CELEX dataset yet.

8.8.2 Conclusion

In this section, it has been shown that it is possible to integrate (restricted) joint-*n*-gram features within HCRF training to improve model accuracy for a G2P task. On the English CELEX database, the performance on the evaluation set could be improved by roughly 5% relative over our best HCRF system without joint-*n*-gram features. In comparison with other approaches in the literature, HCRFs lead to very good results and are comparable with the results achieved by the state-of-the-art joint-*n*-gram approach as presented in [Bisani & Ney 08] and are quite close to the online discriminative training as presented in [Jiampojamarn & Cherry+ 10]. Although the improvements of the joint-*n*-gram features are encouraging, they are computationally expensive and it is thus prohibitive to use them for larger tasks. Thus, they will not be used for the real-life ASR experiments presented in the next chapter.

Chapter 9

HCRFs for ASR

In virtually every state-of-the-art LVCSR system, G2P is applied to generalize beyond a fixed set of words given by a background lexicon. The overall performance of the G2P system has a strong effect on the recognition quality. A number of different methods have been proposed over the years to tackle this task. Typically, generative models based on joint-*n*-grams are used, although some discriminative models have a competitive performance but the training time may be quite large. The authors of [Chen 03, Vozila & Adams+ 03, Bisani & Ney 08, Novak & Dixon+ 12] build upon graphone-based joint-*n*-gram approaches, whereas the details for alignment of graphemes and phonemes, training and decoding differ. To recall, a "graphone" is a blend of the words grapheme and phoneme and describes an *n*-gram approach trained on a sequence of aligned graphemes and phonemes (cf. Chapter 6). The last two methods are available as open source tools [Bisani 08, Novak 11]. A comparison of generative models for the G2P task is presented in [Hahn & Vozila+ 12] as well as in Chapter 6. On the other side, there are discriminative approaches, which have been proposed rather recently, e.g. online discriminative training [Jiampojamarn & Cherry+ 10], which is also available as an open source tool, or methods based on CRFs.

As has been shown in Chapter 4, CRFs have been successfully applied to NLU tasks in various languages. In Chapter 7, CRF training has been extended by various features, particularly techniques to cope with a high number of features, especially for the G2P task. One very important difference between NLU and G2P tasks is that for the latter, there is usually no alignment provided with the training data. To tackle this problem, HCRFs have been introduced in Chapter 8, leading to state-of-the-art performance on an English G2P task.

While discriminative models usually lead to very good results, the training might be quite demanding with respect to computational time and memory consumption. Since typical (generative) G2P systems usually already have a very good performance ($< 10\%$ phoneme error rate), the effort of using discriminative models is usually not spent; at least not for larger tasks. Additionally, these G2P models are usually only evaluated on a textual level and thus without ASR experiments.

Within this work, up until now, CRFs have only been applied to comparatively small data sets within the NLU domain with respect to the number of training samples (cf. Section A.1). Concerning G2P, HCRFs have only been applied for dry runs, i.e. without ASR experiments. But these experiments are important to see if there is a practical advantage for real-life tasks in using discriminative G2P modeling compared to using generative models.

Thus, the application of HCRFs on an LVCSR task and ASR related questions are the focus

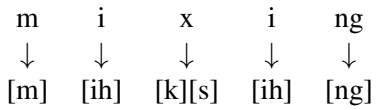


Figure 9.1 Example of a manually aligned word/pronunciation pair from the BEEP pronunciation dictionary, which has been used as background lexicon for AM training and training database for G2P modeling for the English QUAERO system.

of this chapter. The effect of using discriminative G2P modeling based on HCRFs compared to using a generative joint-*n*-gram approach as well as their combination on an English LVCSR task is analyzed. Both methods are shortly recalled in Section 9.1, which is followed by the presentation of the experimental setup in Section 9.2. Besides measuring and comparing the G2P qualities on a textual level in Section 9.3.1, one focus is the performance of LVCSR systems with respect to word error rate. Additionally, we analyze the effect of varying the number of pronunciation variants per word as well as the pronunciation scores on speech recognition performance. The respective results are analyzed and discussed in Sections 9.3.2 - 9.3.4. The chapter is concluded with a summary in Section 9.4. A more condensed version of this work has been presented in [Hahn & Lehnen$^+$ 13].

9.1 G2P Methods

As already presented in e.g. Section 6, there exist a number of methods to tackle the G2P task. For our experimental comparison, we have chosen a generative and a discriminative approach which are recalled shortly in this section including the parameter settings for the experimental setup. As generative model, we have chosen the joint-*n*-gram approach as proposed in [Bisani & Ney 08] since it leads to state-of-the-art results (cp. Section 6) and is available as an open-source toolkit. Naturally, for the discriminative model, we use the HCRF software presented in the previous chapters which is an in-house realization. To tackle the task at hand, an approach needs to be able to handle alignment, training and decoding on comparatively large tasks in reasonable time. An example for a word/pronunciation pair (here presented with a manual alignment) from the chosen training data from the English British English Example Pronunciation (BEEP) pronunciation dictionary as described in detail in Section A.5 is shown in Figure 9.1.

This example illustrates that a one-to-one alignment does not suffice to capture the relations between graphemes and phonemes correctly. Thus, an alignment model needs the capability of modeling some kind of many-to-many alignments since two graphemes need to be aligned to one phoneme and vice versa. Within the sequitur G2P model this is realized by the ability to insert epsilons on both grapheme and phoneme side which can be interpreted as continuations of the former symbol. The HCRF model realizes these links by using a 0-1-2 HMM for the alignment of up to two phonemes to one grapheme and start/continue tags for multiple graphemes aligned to one phoneme, as described in Sections 8.2.3 and 8.6 respectively.

9.1.1 Generative Approach: Joint-n-Gram Model

Models based on joint-n-grams usually rely on graphone sequences $\mathbf{q}$, which are defined as aligned units of graphemes and phonemes, resulting in the following probability decomposition:

$$Pr(\mathbf{g},\boldsymbol{\varphi}) = Pr(\mathbf{q}) = \prod_{i=1}^{N} Pr(q_i|q_{i-1},\ldots,q_1) \tag{9.1}$$

The resulting n-gram model as proposed in [Bisani & Ney 08] is defined as

$$p(\mathbf{g},\boldsymbol{\varphi}) = \sum_{\mathbf{q}\in S(\mathbf{g},\boldsymbol{\varphi})} p(\mathbf{q}) \tag{9.2}$$

$$= \sum_{\mathbf{q}\in S(\mathbf{g},\boldsymbol{\varphi})} \prod_{i=1}^{|\mathbf{q}|} p(q_i|q_{i-1},\ldots,q_{i-M+1}) \tag{9.3}$$

Here, $S(\mathbf{g},\boldsymbol{\varphi})$ denotes the set of all co-segmentations of $\mathbf{g}$ and $\boldsymbol{\varphi}$ whereas M denotes the LM model order. Training of the model is performed using maximum likelihood EM training. For decoding, a maximum approximation is applied for the possibly non-unique segmentation into graphones (cf. Section 6.1.4 for more details).

The joint-n-gram model has been trained using the open-source toolkit Sequitur [Bisani 08]. There are basically two parameters which control the quality of the model: a length restriction on the graphones (graphemes and phonemes can be restricted separately) and the n-gram order. For the graphones, we use the setting which has been reported to work best for English tasks, namely to allow the use of graphones of length one, whereas the grapheme or phoneme may be empty. The performance on the development set converges at $n = 8$, i.e. the G2P approach works best on the given data with an graphone-level eight-gram model. In search, the maximum approximation is applied.

9.1.2 Discriminative Approach: HCRFs

Compared to linear chain CRFs as introduced in [Lafferty & McCallum$^+$ 01], HCRFs additionally model an alignment between a source sequence (graphemes $\mathbf{g} = g_1,\ldots,g_L$) and a target sequence (phonemes $\boldsymbol{\varphi} = \varphi_1,\ldots,\varphi_N$), which is needed for G2P tasks. The alignment is integrated via a hidden variable. HCRFs, e.g. [Quattoni & Wang$^+$ 07, Koo & Collins 05], and hidden dynamic conditional random fields (HDCRFs) [Yu & Lam 08] have been proposed in the literature. Our approach is similar to the latter one, where a sum over all possible alignments a_1^L is additionally introduced in training:

$$p(\boldsymbol{\varphi}|\mathbf{g}) = p_{\lambda_1^M}(\varphi_1^N|g_1^L) = \frac{\sum_{a_1^L} \exp H(\varphi_1^N, a_1^L, g_1^L)}{\sum_{\tilde{a}_1^L}\sum_{\tilde{\varphi}_1^N} \exp H(\tilde{\varphi}_1^N, \tilde{a}_1^L, g_1^L)} \tag{9.4}$$

with

$$H(\varphi_1^N, a_1^L, g_1^L) = \left(\sum_{n=1}^{N}\sum_{m=1}^{M} \lambda_m \cdot h_m(\varphi_{n-1}, \varphi_n, a_1^L, g_1^L)\right) \tag{9.5}$$

$H(\varphi_1^N, a_1^L, g_1^L)$ represents position dependent, binary feature functions $h_m(\varphi_{n-1}, \varphi_n, a_1^L, g_1^L)$. The maximization of the conditional log-likelihood is used as training criterion for the feature weights λ_1^M over a given training dataset. The decision criterion is given by the maximization of the sentence-wise probability $p(\varphi_1^N | g_1^L)$, i.e. a maximum approximation is applied as for the joint-n-gram approach. To cope with the high computational complexity, certain restrictions are applied. Details about our implementation are given in [Lehnen & Hahn$^+$ 12]. It should be noted that within the HCRF and the Sequitur approach, an alignment respectively a segmentation of the data is implicitly and additionally learned.

For the HCRF model, we used lexical and source-n-gram features in a windows of $-5 \ldots 5$ around the current grapheme as well as the powerful bigram features on phoneme side. Additionally, a (Gaussian) prior has been introduced for smoothing. The joint-n-gram feature as introduced in Section 8.8.1 has not been used due to the high computational complexity. On the BEEP lexicon, there are roughly 157M features. Due to this large amount of features, feature selection methods have been applied (e.g. elastic net [Lavergne & Cappé$^+$ 10] and feature count cut-off as presented in Section 7.3) resulting in 28M active features, i.e. features with non-zero weight. We trained the model until convergence after 50 RProp iterations. Due to the non-convexity of the training criterion, the lexical features (φ_n, g_{a_l}) have been initialized with IBM-1 scores as described in Section 8.6.

9.2 Experimental Setup

In this section, the training schedule of the ASR system is presented as well as the strategy to integrate G2P into the ASR system. The various data sources used for training and testing are also introduced, whereas the experimental results will be presented in the following section.

9.2.1 ASR System

The used two-pass ASR system is based upon the English QUAERO system as described in [Sundermeyer & Nußbaum-Thom$^+$ 11]. As features, MFCCs were appended by a voicedness feature and phone-posterior-based features estimated using a MLP. More precisely, we usee hierarchical multiple RASTA (HMRASTA) bottleneck features. The acoustic model itself is based on across-word triphone states represented by left-to-right three-state Hidden Markov Models. For speaker normalization, we applied VTLN on the feature vectors. C-MLLR has been used as speaker adaptation technique in training and recognition. For all presented results, we used minimum phone error (MPE) as discriminative training criterion. A pruned four-gram LM smoothed by modified Kneser-Ney discounting has been applied. This LM has been trained on approx. 3B words in various corpora, which have been linearly interpolated to optimize perplexity on a holdout data set.

9.2.2 Integrating G2P and ASR

For the ASR experiments, we first fixed a recognition vocabulary of 150K words (more precisely: 150.035), as usual based on count statistics from text data available for the task at

Table 9.1 Statistics about the overlap between the pronunciations of the two G2P models and the BEEP background lexicon. Both methods can more or less replicate the Beep pronunciations, but only 69.9% of the HCRF pronunciations are also in Seq; the other way around it is 70.9%.

	vocabulary	# pronunciation variants	ratio of pronunciation variants [%] in HCRF	in Seq
BEEP	69,956	76,318	98.4	96.6
HCRF	145,385	177,986	100.0	69.9
Seq	145,385	179,656	70.9	100.0

hand. Pronunciations for 5K regular abbreviations have been added via a rule-based approach (spelling of single letters). For the remaining 145K words, we applied both G2P methods with the following setting: for each word, up to four pronunciation variants are generated. A variant is added to the lexicon, if it has a posterior confidence score ≥ 0.2. In several evaluations, we have found that this recipe leads to good performance on ASR tasks. For the time being, we do not use pronunciation weights. A comparison of the overlap between the two resulting lexica and the background lexicon is given in Table 9.1.

Concerning the left part of the table, from the 145K words for the recognition lexicon, roughly 70K are within the BEEP lexicon (as shown in the first line of the table), meaning that at least for the remaining 75K words, a G2P model has to be applied, which is for more than 50% of all words. Line two and three show the statistics for the HCRF and the sequitur system applied to all words of the recognition vocabulary. As can be seen in the right part of the table, both G2P methods can more or less replicate the pronunciation variants from the background lexicon, which was to be expected since the models have been trained on exactly that data. Interestingly, the pronunciations generated by the G2P models for words which are not part of the BEEP lexicon do differ by roughly 30%. Thus, there is an effect on ASR performance to be expected.

9.3 Experimental Results

Various experiments have been performed to measure the effect of G2P modeling on LVCSR performance. First, we will analyze and compare the two G2P systems which have been used within the ASR experiments. As error measure for these systems, we use the standard PER and WER. As already defined in Section 6.2.1, the PER is defined as the ratio of insertions, deletions and substitutions of a Levenshtein alignment between a hypothesis and a reference phoneme sequence and the reference length. If there are multiple references, the alignment is done w.r.t. all references and the one with the least errors is chosen. The WER is defined as the number of wrongly recognized pronunciations w.r.t. the total number of reference pronunciations. Second, we will analyze the effect of three factors which influence the modeling of the lexicon: First, the G2P strategy itself, second, the number of pronunciation variants and third, the kind of pronunciation scores. For these ASR experiments, we use the well-known WER as measurement.

Table 9.2 Phoneme Error Rates (PER) and Word Error Rates (WER) on the BEEP development set for the Sequitur and the HCRF G2P system.

approach	PER[%]				WER[%]
	sub	del	ins	total	
Seq	1.0	0.4	0.4	1.7	9.0
HCRF	1.2	0.5	0.3	2.1	11.6

9.3.1 G2P Systems

The results of the two tested G2P systems on the BEEP development set are presented in Table 9.2. This set comprises roughly 10K words corresponding to 4% of the total database and has been set aside from the training material as described in Section A.5.

With a PER of 1.7%, the Sequitur approach leads to better results than the HCRF approach with 2.1% PER. The deletion-insertion ratio is quite even for the Sequitur system, whereas there is a small bias towards deletions in the HCRF system. The number of substitutions for the HCRF system is significantly higher than for the Sequitur approach. The overall performance of both methods on the BEEP lexicon is quite good compared to other English G2P tasks (cp. e.g. Table 6.1 or Table 6.2).

9.3.2 LVCSR - Varying G2P Strategy

We performed all reported ASR experiments on the English QUAERO data as described in Section A.5, which is a state-of-the-art task. Concerning the LVCSR experiments, we stuck to the following procedure: the vocabulary for training and recognition as well as the acoustic and language modeling data has been fixed and is the same for all experiments. We only changed the way of generating pronunciations. Since the G2P model is also needed in training of the AM, we did a complete training from scratch for various ASR systems. Additionally, we always use pronunciation weights calculated on the training alignment via a forced alignment as presented in [Gollan & Ney 08] for recognition. The corresponding formulation is given in Equation 9.6.

$$p_{\mathrm{ps}}(\boldsymbol{\varphi}|\mathbf{g}) := \frac{N(\boldsymbol{\varphi},\mathbf{g})}{\lambda + N(\mathbf{g})} + \frac{\lambda}{\lambda + N(\mathbf{g})} p_{\mathrm{ed}}(\boldsymbol{\varphi}|\mathbf{g}) \tag{9.6}$$

Here, the originally equally distributed pronunciation weights p_{ed} are weighted by the counts of observed pronunciations $N(\boldsymbol{\varphi},\mathbf{g})$ normalized by the corresponding word counts $N(\mathbf{g})$ and a balancing parameter λ which has to be adjusted empirically. In our experiments, $\lambda = 10$ leads to good results. Although this method leads to small improvements, the disadvantage is that only words which have been observed in training will get pronunciation scores. Words which are not observed in the training data have equally distributed pronunciation weights.

We also tried to use pronunciation weights in training to better guide the alignment process. Therefore, we performed a kind of second iteration AM training, where we used the previous AM for a forced alignment between the training audio data and the reference transcription. The pronunciation weights calculated using the above formula have been used to augment the

Table 9.3 ASR results on various English QUAERO development and evaluation sets with varying G2P strategies. The first line represents the baseline system, where the Sequitur G2P model was only used iff the respective word was not in the Beep lexicon. This "hierarchical" lookup is denoted by "→". The "∪"-symbol denotes a merge of the models' hypotheses.

system number	pronunciation lookup	WER[%] dev10	eval10	eval10	eval11	eval11	eval12
1	Beep → Seq	16.5	16.4	16.4	21.8	21.7	18.7
2	Seq	16.6	16.4	16.3	21.7	21.6	18.4
3	Seq ∪ Beep	16.4	16.3	16.2	21.5	21.3	18.3
4	HCRF	16.4	16.4	16.2	21.3	21.2	18.3
5	HCRF ∪ Beep	16.4	16.2	16.1	**21.2**	21.1	18.4
6	HCRF ∪ Seq ∪ Beep	16.3	**16.2**	16.2	21.2	21.0	**18.1**
7	Beep → HCRF	16.3	**16.3**	16.1	**21.3**	21.0	**18.0**

training lexicon and using this augmented lexicon we started an AM training from scratch. This procedure did not lead to improvements. Most probably, the alignment is already robust enough without the pronunciation weights.

The experimental results for the two tested G2P strategies and possible combinations are presented in Table 9.3 for three pairs of data sets, whereas the left set has been used for parameter optimization and the right one for testing. We have chosen three different evaluation sets to ensure the significants of our findings, since we are aware that only varying the G2P strategy might lead to effects which are barely notable.

As baseline system, we use the BEEP lexicon for pronunciation lookup and only if the respective word is not within the lexicon, we use the Sequitur G2P strategy (system 1 in the table). This hierarchical kind of lookup is denoted by "→" in the table and correspond to our standard setup for more or less any ASR systems, i.e. first a lookup in the background lexicon and application of Sequitur G2P only for misses. To get contrastive results, we have used all meaningful (non-hierarchical) combinations of HCRF, Sequitur and the BEEP lexicon to retrieve pronunciations. System 2 uses just the Sequitur G2P strategy without lookup in the BEEP lexicon. For system 3, the pronunciation lookup has been performed with the BEEP lexicon and the Sequitur G2P system and both outputs have been merged, denoted by "∪". The two following systems use the HCRF system instead of Sequitur. A combination of all available knowledge sources is the basis for system 6. For system 7, we just replaced Sequitur by HCRF in the baseline system.

Although there is no clear best system across the three test cases, systems relying on HCRFs as G2P method seem to outperform systems based on Sequitur, although the HCRF G2P model performs worse on text data (cf. Table 9.2). The best systems 5, 6 and 7 (denoted in bold) lead to a gain in performance between 1–3% relative over the baseline system. 0.1% WER corresponds to 35–45 word errors depending on the data set. For the exact sizes, cf. Table A.6. Note that we only changed the G2P strategy and nothing else. Thus, HCRFs seem to be able to

Table 9.4 Number of pronunciation variants for various G2P strategies for the English QUAERO task. The first line represents the baseline system, where the Sequitur G2P model was only used iff the respective word was not in the BEEP lexicon. This "hierarchical" lookup is denoted by "$\rightarrow$". The "$\cup$"-symbol denotes a merge of the models' hypotheses.

system number	pronunciation lookup	# pronunciation variants
1	Beep $\rightarrow$ Seq	181.604
2	Seq	184.313
3	Seq $\cup$ Beep	189.303
4	HCRF	182.644
5	HCRF $\cup$ Beep	183.882
6	HCRF $\cup$ Seq $\cup$ Beep	239.150
7	Beep $\rightarrow$ HCRF	178.588

generalize very well and are suitable for LVCSR systems.

In Table 9.4, the numbers of pronunciations within the recognition lexicon for the various G2P lookup strategies are presented.

Except for system 6, the numbers do not change significantly. This was to be expected, since it has already been reported in Table 9.1 that the difference between the pronunciations generated by Sequitur and HCRF are quite high, although the numbers per system are quite similar. Interestingly, a combination of Sequitur and HCRF pronunciations did not lead to much better or worse results, although the number of pronunciations is much higher than for any of the other systems.

9.3.3 LVCSR - Varying Number of Pronunciation Variants

Within another set of experiments, we wanted to analyze the effect of varying the number of pronunciation variants and also the use of G2P confidence scores as pronunciation scores. Therefore, we have chosen the two ASR systems which rely only on a G2P model for pronunciation modeling (cf. systems 2 and 4 in Table 9.3). We optimized the number of (fixed) pronunciations per word on the dev10 set without using the confidences for thresholding. The idea behind this experiment is to analyze if one of the two G2P methods can benefit more from a higher number of pronunciation variants than the other. Additionally, since both methods are capable of providing confidence scores, we wanted to see if these could be used as pronunciation scores. The respective results are presented in Table 9.5.

For both, the HCRF and the Sequitur approach, three experiments have been carried out: one without using pronunciation scores at all, one where the method's confidence scores are used as pronunciation scores and one where the pronunciation scores are calculated using the alignment on the training data as presented in Equation 9.6. Independent of the type of pronunciation score used, the HCRF model outperforms the Sequitur model. Within each method, the pronunciation scores calculated on the alignment of the training data outperform the system's confidence

Table 9.5 ASR results for Sequitur and HCRF G2P modeling with fixed number of pronunciation variants and varying types of pronunciation scores on the English QUAERO sets. The (fixed) number of pronunciations per word has been optimized on the dev10 development set and the optimal number varies with the type of pronunciation score.

	setup pronunciation scores from	# pronunciations/ word	WER[%] dev10	eval10	eval10	eval11
HCRF	none	2	17.6	17.4	17.4	22.7
	G2P confidences	4	16.7	16.6	16.5	21.7
	train alignment	5	16.3	**16.4**	16.3	**21.4**
Seq	none	2	18.0	18.0	18.0	23.1
	G2P confidences	3	16.7	17.0	16.9	22.3
	train alignment	3	16.7	**16.7**	16.7	**21.8**

scores. But using confidence scores leads to better results than not using any pronunciation weights. Additionally, if no pronunciation scores are used, more than 2 pronunciations per word lead to worse systems whereas the optimal number of pronunciations per word is higher if pronunciation scores are used. Using pronunciation scores leads to a gain in all cases. The number of pronunciation variants leading to optimal results when pronunciation scores are used is higher for the HCRF method than for the Sequitur approach, which can be interpreted in a way that the ASR system benefits more from the HCRF variants.

9.3.4 LVCSR - Varying Pronunciation Scores

With a last set of experiments, we wanted to overcome one drawback when using the training alignment as the only source for pronunciation scores: this is only possible for words (more precisely: pronunciations) which occur in the acoustic training data. We want to be able to assign pronunciation scores to all words in the recognition lexicon. Additionally, we wanted to verify the gain by using pronunciation scores without varying the number of pronunciations. As baseline systems, we again use the systems based on Sequitur and HCRFs only, i.e. without BEEP lookup. The results are presented in Table 9.6.

Whereas systems 2 and 6 show the baseline results which are taken from Table 9.3, systems 1 and 5 show results without using pronunciation scores at all, which means that all variants are weighted equally. To include pronunciation scores apparently helps and leads to small but consistent improvements of about 1% relative. When raw G2P posterior scores are used as pronunciation scores (systems 3 and 7), the quality of the ASR system drops. The results are even worse than when using no pronunciation scores at all. Even a hierarchical combination of the G2P system with the scores from the training alignment does not help. Here, the respective G2P system is only used when the pronunciation variant has not been observed in the training data and thus there would be no pronunciation score otherwise (systems 4 and 8). It should be noted that the posterior scores are always normalized per word, e.g. across all pronunciation

Table 9.6 Results for the ASR systems based on Sequitur and HCRF G2P modeling only. Here, only the pronunciation scores have been varied. For the "mix" lines, the G2P system's confidence score has been used as pronunciation score iff the pronunciation did not occur in the training alignment.

system number	setup		WER[%]			
		pron scores from	dev10	eval10	eval10	eval11
1	HCRF	none	16.6	16.4	16.4	21.7
2		train alignment	16.4	**16.4**	16.2	**21.3**
3		G2P confidences	16.6	16.5	16.3	21.6
4		mix	16.4	**16.4**	16.2	**21.3**
5	Seq	none	16.7	16.5	16.5	21.9
6		train alignment	16.6	16.4	16.3	21.7
7		G2P confidences	16.9	16.8	16.6	22.1
8		mix	16.6	16.4	16.3	21.8

variants per word. Thus, the posterior scores of the G2P systems do not seem to be of any help with respect to the weighting of pronunciation variants.

9.4 Conclusion

In this section, we have shown that G2P modeling using HCRFs can outperform a generative Sequitur G2P model within LVCSR experiments, even if the PER of the HCRF model on text data is worse than the Sequitur approach. Improvements of 1–3% could be achieved across a number of test sets from the English QUAERO tasks. To include pronunciation variants is also helpful, especially if pronunciation scores are used. We could also verify that pronunciation weights calculated on the training alignment improve performance. To include posterior scores from G2P systems as pronunciation weights, even in a supplemental manner for variants which are not seen in training, does not improve performance. If the number of pronunciations per word is fixed and not dependent on the G2P confidence score, the use of G2P confidence scores can improve on not using any confidence scores at all, but is still worse than using scores calculated on the training alignment.

It might be worth to analyze the effect of combining the Sequitur/HCRF confidence scores with acoustic scores as pronunciation weights or to vary the number of pronunciation variants based on confidence scores. Additionally, it might be worth to apply system combination to the ASR systems with varying G2P methods.

Chapter 10

Scientific Contributions and Conclusion

The comparison of various state-of-the-art methods to tackle monotone string-to-string translation tasks, more precisely concept tagging and grapheme-to-phoneme conversion, and to improve these results using models based on conditional random fields (CRFs) has been the main focus of this work. In particular, the following contributions are contained in this work covering various aspects of and insights into solving monotone string-to-string translation tasks:

A comparison of state-of-the-art methods for concept tagging

Extensive comparisons have been derived for both the concept tagging and the grapheme-to-phoneme conversion task. For concept tagging, six state-of-the-art methods (FST, DBN, SMT, SVM, MEMM, CRF) have been trained and compared on NLU tasks in three languages, namely French, Polish, and Italian. Additionally, these tasks have varying complexity and vocabulary sizes, which supports robust and transferable results. The extraction of attribute names only and in combinations with attribute values have been compared. Manual transcriptions and speech input has been considered as input for each of the methods. Overall, CRF have been found to lead to the best results across languages and input modalities. On the well-known French MEDIA corpus, a CER of 10.6% resp. 12.6%, if attribute value extraction is considered additionally to attribute name extraction, could be achieved on the evaluation set using manual transcriptions as input. This corresponds to a relative reduction of approx. 35% w.r.t. results in the literature. With automatic transcriptions, the comparable figures are 23.8% and 27.3%. Thus, when attribute values are additionally extracted, the CER raises by approx. 17–27% relatively. For Polish and Italian, there are no comparable figures available by other groups yet, since the corpora have been collected only recently. But a CER of 24.7% on the evaluation set for Polish text input, attribute name and value extraction, and 21.8% CER for the comparable figure in Italian seem to be a good start. The recognition errors introduced by ASR systems are a particular challenge. Especially for French and Polish, the CER is more than doubled when the erroneous input is used. Many errors are due to background noises and multiple voices, or mispronunciations and OOV words. In general, discriminative methods seem to lead to more robust results on speech input than generative approaches. Concerning CRF, especially word-part features seem to increase the robustness. The findings have been published in [Hahn & Dinarelli$^+$ 11].

A comparison of state-of-the-art methods for grapheme-to-phoneme conversion

Concerning the G2P task, the methods proposed in [Kneser 00, Chen 03, Vozila & Adams+ 03, Bisani & Ney 08, Novak 11] have been trained and compared on the medium-sized English CELEX task as well as on large, state-of-the-art Western-European pronunciation dictionaries in English, German, French, Italian and Dutch. Overall, methods based on [Bisani & Ney 08] and [Novak 11] lead to the best results. Additionally, we compared these models w.r.t. their effect on ASR performance within contemporary LVCSR systems. With a cheating experiment where OOVs have been added to the recognition lexicons, it could be shown that improved G2P modeling can be measured within ASR systems even over a highly competitive baseline. Additionally, using n-best pronunciations seems to help for most languages. In any case, even when improving G2P modeling does not improve ASR performance significantly, improving G2P is always beneficial for end user customization. The respective findings have been published in [Hahn & Vozila+ 12].

Application of ROVER system combination

Due to the principal differences between the various modeling approaches described in the previous paragraph (e.g. generative versus discriminative), system combination results revealed possible synergetic effects for the concept tagging tasks. It was possible to improve the single-best system for all languages and manual transcriptions as input from 1–12%. For speech input, the results vary, mainly due to the large gap between the CRF systems and the second best models. For G2P, system combination did not lead to statistically significant improvements, mainly due to the already very low error rates and the similarity of the applied approaches. The ROVER results are also published and discussed within the publications cited in the previous paragraph as well as in [Hahn & Lehnen+ 08a] for concept tagging.

Application of CRFs to concept tagging

A CRF framework has been developed as a module to the RWTH ASR speech recognition engine. With this in-house CRF realization, all the reported experimental results on concept tagging and G2P tasks have been obtained. By selecting and tuning features, the best published results on the presented tagging corpora could be achieved. One additional improvement to the CRFs could be obtained by introducing a margin term to the training criterion leading to improvements of 4–8% relative. With respect to attribute value extraction, it was possible to improve the standard rule-based extraction by combining it with a statistical approach based on CRFs. A CER reduction by 2–7% relative across languages could be achieved, whereas the improvement is slightly larger for manual transcriptions as input than speech. Here, a second restricted CRF model has been trained on the attribute names and words as input sequence and the attribute values as output sequence. For each attribute name, only one attribute value has been allowed to be hypothesized. The combination of both methods has been performed as follows: After an error analysis of both approaches per attribute name, the approach with less errors has been chosen to be applied for the respective attribute name. The respective findings are published in [Hahn & Dinarelli+ 11], whereas the introduction of the margin term is discussed in more detail in [Hahn & Lehnen+ 09].

Application of (H)CRFs to grapheme-to-phoneme conversion

Since the requirement for this task differs from the concept tagging task, feature functions and methods to filter respectively reduce the number of active features had to be derived. We found that dropping the prefix, suffix and capitalization features, which do not have a meaning within the G2P task, and including source-side *n*-grams lead to good results. The filtering is done using the elastic net (EN) approach amongst others, reducing the number of active features to below 1% of the features of the original, full model without loss of performance. It is also possible to replace the costly bigram feature with an *n*-gram LM on target side without loosing performance. The respective findings are published in [Hahn & Lehnen$^+$ 11].

Additionally, an alignment has to be derived between graphemes and phonemes, which was not necessary for the concept tagging task. Various methods have been explored starting from the use of a pre-computed alignment produced by an external tool to the successful integration of the alignment as a hidden variable, referred to as hidden conditional random field (HCRF). It has been shown that it is possible to integrate (restricted) joint-*n*-gram features within HCRFs training to improve model accuracy for the G2P task. On the English CELEX database, the performance on the evaluation set could be improved by roughly 5% relative over our best HCRF system without joint-*n*-gram features. In comparison with other approaches in the literature, HCRFs including these features lead to very good results and are comparable with the results achieved by the state-of-the-art joint-*n*-gram approach as presented in [Bisani & Ney 08] and are quite close to the online discriminative training as presented in [Jiampojamarn & Cherry$^+$ 10]. The respective results have been published in [Lehnen & Hahn$^+$ 11a, Lehnen & Hahn$^+$ 11b]. Using all of these improvements, it was possible to obtain state-of-the-art results for G2P tasks using CRFs.

Investigations on the effect of using HCRF G2P within an LVCSR system

G2P systems by itself are rarely deployed respectively used in practical systems. To get an idea of the effect of utilizing a HCRF-based G2P approach instead of and in combination with a standard joint-*n*-gram based system, various LVCSR systems have been trained where different G2P systems are used to derive pronunciations for OOVs. Even if the phoneme error rate (PER) for the HCRF system leads to worse results than the Sequitur approach, it was possible to improve LVCSR performance by 1–3% relative by just replacing the G2P strategy across a number of English QUAERO tasks. Pronunciation variants have been found to be always beneficial, especially when combined with pronunciation scores derived from the training alignment. The respective findings are published in [Hahn & Lehnen$^+$ 13].

Chapter 11

Outlook

Although the performance of CRFs on NLU tasks is already pretty good compared to other approaches, there seems to be some room for improvement when looking at system combination results. It might be worth to have a deeper look into the difference of the various approaches and to add features to the CRF approach which model characteristics of other methods like FSTs which are not yet covered by the currently used CRF feature sets.

The performance might also improve if longer contexts on target side (i.e. trigrams, fourgrams, ...) are used. This would also require techniques to efficiently prune the FSA representing $p_{\Lambda,CRF}(t_1^N|g_1^N)$ since the memory complexity would be too high otherwise. Overall, improvements in the central processing unit (CPU) and memory footprint might lead to a successful application of CRFs to the general task of machine translation (cf. e.g. [Lavergne & Crego$^+$ 11]).

Additionally, w.r.t. statistical attribute value extraction, it might be worth to apply factorial CRFs instead of the hierarchical approach presented in this work [Sutton & McCallum$^+$ 07]. This might not propagate search errors made in the attribute name extraction step and thus improve performance on both the attribute name and the attribute value extraction.

Concerning the application of CRFs to the G2P tasks, the performance is comparable to state-of-the-art approaches. It might as well as for the NLU task be improved by reducing the memory and CPU footprint which would allow for less drastic feature selection and pruning strategies. It has been shown in the literature that the addition of joint-n-gram with longer contexts than two on target side might lead to better results [Jiampojamarn & Cherry$^+$ 10].

Concerning the application of HCRFs for ASR, it might be worth to analyze the effect of combining the Sequitur/HCRF confidence scores with acoustic scores as pronunciation weights or to vary the number of pronunciation variants based on confidence scores. Additionally, it might be worth to apply system combination to the ASR systems using varying G2P methods.

Appendix A

Corpora and Systems

In this chapter, the various datasets which are used for the experimental results reported in this work are presented. Since the statistics are needed in various sections, they are collected here for reference. Additionally to the NLU and G2P corpora, the statistics for the English QUAERO ASR system are also presented in this chapter.

A.1 LUNA NLU Corpora

Within the LUNA project[1], which was an EU sixth framework funded, Information Society Technologies (IST) Integrated Project (IP) project (FP6-033549) with a duration of three years (2006–2009), three different telephone speech corpora in three different languages (French, Polish, and Italian) have been collected respectively improved. The corpora have been annotated on various levels, which include manual transcription, automatically obtained POS tags, basic constituents based on the POS tags called chunks, domain specific concept attribute and value tags, predicate structure, co-reference/anaphora and dialog acts as the highest level of annotation. More documentation on the various levels of annotation can be found within the publicly available deliverable D1.3 on the project's homepage [Rodriguez & Riccardi$^+$ 07] and is as well shipped with the corpora. Additionally to the manual transcriptions of the audio data, ASR output is also provided to test NLU methods for robustness, also referred to as SLU.

One benefit in having three different languages and domains is that evaluations make it possible to compare performances on the manually annotated data with the annotations obtained with ASR hypotheses and to establish and to observe some trends consistent across the corpora. In this work, we are only concerned with concept tagging, which is a low-level step within any SLU respectively dialog system. As input, we use the transcription (either manual or by the ASR system) and as output we want to generate concept tags. Thus, the corpora descriptions and statistics in this section only refer to these two annotation levels. The French MEDIA corpus is publicly available with manual transcriptions and annotations in terms of concept tags and values. It consists of human-machine dialogues collected with a Wizard of Oz procedure involving selected speakers. The other two corpora were specifically acquired, manually transcribed and annotated with semantic information for the LUNA project. The Polish corpus consists of human-human conversations recorded in the call center of the Polish Warsaw transportation system while the Italian corpus consists of dialogues of a help-desk application

[1] http://www.ist-luna.eu/

Table A.1 Statistics of the training, development and evaluation LUNA SLU corpora as used for all experiments concerned with concept tagging.

		training		development		evaluation	
		words	concepts	words	concepts	words	concepts
French	# sentences	**12,908**		**1,259**		**3,005**	
	# tokens	94,466	43,078	10,849	4,705	25,606	11,383
	# NULL tokens	32,580	11,442	4,157	1,372	9,040	2,999
	vocabulary	2,210	99	838	66	1,276	78
	# singletons	798	16	338	4	494	10
	OOV rate [%]	–	–	1.33	0.02	1.39	0.04
Polish	# sentences	**8,341**		**2,053**		**2,081**	
	# tokens	53,418	28,157	13,405	7,160	13,806	7,490
	# NULL tokens	21,973	9,811	5,680	2,384	5,743	2,486
	vocabulary	4,081	195	2,028	157	2,057	159
	# singletons	1,818	19	1,119	23	1,113	28
	OOV rate [%]	–	–	4.95	0.13	4.96	0.11
Italian	# sentences	**3,171**		**387**		**634**	
	# tokens	30,470	14,683	3,764	1,818	6,436	3,057
	# NULL tokens	15,233	5,872	1,893	723	3,287	1,242
	vocabulary	2,386	43	777	39	1,059	39
	# singletons	1,140	0	417	4	537	3
	OOV rate [%]	–	–	4.22	0.06	3.68	0.00

in which the employees of the Consorzio per il Sistema Informativo Piemonte (CSI), a public regional institution, seek advice on problems related to their computers. The characteristics of the three corpora are summarized in Table A.1.

Since we only want to perform concept tagging on word sequences uttered by (human) users, only these user turns have been used to train and test the models. No filtering of turns has been carried out, since we want to get as close as possible to real-life applications. Thus, some turns may contain just the NULL tag indicating chunks that do not convey a meaning relevant for the application domain. In the following subsections, each of the corpora is described in detail. As can be seen in the aforementioned table, the percentage of NULL tokens is quite high within all three corpora. This can in some way be compared to the portion of silence in ASR systems.

A.1.0.1 The French MEDIA corpus

This corpus was collected in the French Media/Evalda project in the domain of negotiation of tourist services [Bonneau-Maynard & Rosset$^+$ 05]. It is divided into three parts: a training set consisting of 13k sentences, a development set (1.3k sentences) and an evaluation set (3.5k sentences). There are 99 different attribute name tags ranging from simple date and time expressions to more complex ones like co-references. Note that the concept names are also in French and not in English as for the other two corpora. One typical example sentence from the

je veux une chambre double pour deux personnes

I would like a double-bed room for two persons

(a) Manual transcription with English translation for reference (not included in the corpus).

null{je veux} **nombre-chambre**{une} **chambre-type**{chambre double}
sejour-nbPersonne{pour deux personnes}

null{I would like} **number-room**{a} **room-type**{double-bed room}
stay-numberPersons{for two persons}

(b) Attribute name tags for the example sentence.

nombre-chambre[1] **chambre-type**[double] **sejour-nbPersonne**[2]

number-room[1] **room-type**[double] **stay-numberPersons**[2]

(c) Final attribute name and attribute value sequence for the example sentence.

+nombre-chambre[1] **+chambre-type**[double] **+sejour-nbPersonne**[2]

+number-room[1] **+room-type**[double] **+stay-numberPersons**[2]

(d) Final attribute name and attribute value sequence for the example sentence including modes.

Figure A.1 Example sentence from the French MEDIA corpus with attribute name and value concept tags. The tags are provided in French only; the English translation is just given for reference.

MEDIA training corpus dealing with the reservation of a hotel room is presented in Figure A.1.

The manual transcription of the sentence is given in Figure A.1 a. Note that the English translation is given for reference only and is not part of the corpus. In Figure A.1 b, the annotation of the sentence with concepts (attribute names) is presented. This annotation essentially segments the input sentence into chunks. In Figure A.1 c, the final concept sequence including attribute values is shown, which the SLU system is expected to hypothesize by placing the values between brackets. Additionally, since the NULL concept does not convey any meaning, it is deleted.

The MEDIA corpus also includes annotations, called *specifiers*, about certain relations between concept names and values that are semantic structures. Furthermore, other annotations are included to represent the major speech act of a sentence, like assertion, negation and request [Bonneau-Maynard & Ayache$^+$ 06]. They define the so called *mode* of a sentence. These annotations refer to more complex semantic relations than attribute name/value pairs and are not considered in the experiments described in this work. Not considering them corresponds to

et bien non parce que s' il n' y a pas de
parking surveillé et de restaurant

okay, no because there is no
guarded parking and no restaurant

(a) Manual transcription with English translation for reference (not included in the corpus).

null{et bien} **+reponse**{non} **+connectProp**{parce que}
null{s' il n' y a} **-hotel-parking**{pas de parking surveillé}
-hotel-services{et de restaurant}

(b) Attribute name tags for this example including modes.

+reponse[non] **+connectProp**[explique] **-parking**[prive] **-services**[restaurant]

+answer[no] **+connectingClause**[explanation] **-parking**[private] **-services**[restaurant]

(c) Final attribute name and attribute value sequence for this example including modes.

Figure A.2 Example sentence from the French MEDIA corpus with attribute name and value concept tags illustrating the two modes "+" and "-", which are prefixed to the concept tag names; the English translation is just given for reference.

operate in the so-called *relaxed simplified* condition defined in the MEDIA project. Within this condition, only two modes have to be distinguished ("+" and "-"). Thus, the reported experiments can be directly compared to the results from the MEDIA project. We do not incorporate any special technique to deal with the two modes, but we include them as prefix of the attribute names. Since not all concept tags do occur in both modes within the corpus and the NULL tag does not have a mode, this approach leads to a total of 99 different concept tags. Since the example sentence in Figure A.1 d does only contain the "+" mode, another example containing both modes is given in Figure A.2. Again, the English translation is only given for clarity.

A.1.0.2 The Polish LUNA corpus

The data for the Polish corpus are human-human dialogues collected at the Warsaw Transportation call-center [Marasek & Gubrynowicz 08, Mykowiecka & Marasek$^+$ 09]. This corpus covers the domain of transportation information like e.g. transportation routes, itinerary, stops, or fare reductions. Three subsets have been created using the available data: a training set comprising approx. 8k sentences, a development and an evaluation set containing roughly 2k sentences each. It is the first SLU database for Polish and from the three corpora presented in this paper the most complex one. The number of different annotated concepts is close to 200,

(jestem) na Polnej$_{adj,fem,loc}$/Dąbrowskiego$_{adj,masc,gen}$
(I am) on Polna Street / Dąbrowskiego Street

(jadę) z Polnej$_{adj,fem,loc}$ /Dąbrowskiego$_{adj,masc,gen}$
(I am coming) from Polna Street / Dąbrowskiego Street

(jadę) na Polna$_{adj,fem,acc}$ / Dąbrowskiego$_{adj,masc,gen}$
(I am going) to Polna Street / Dąbrowskiego Street

Figure A.3 Example sentences from the Polish SLU corpus showing the complexity of the language due to inflection and relatively free word order. In theses phrases there are three different concepts describing places: **location_str**, **source_str** and **goal_str**. Here, str is an abbreviations for street.

the largest in the three corpora. Furthermore, many concepts are closely related. The SLU task is particularly difficult in this case because Polish is an inflectional language with a relatively free word order. An example of different types of inflection for Polish location names is given in Figure A.3.

A typical sentence from the corpus including the attribute name and value annotation is given in Figure A.4.

A.1.0.3 The Italian LUNA corpus

The application domain of the Italian corpus [Dinarelli & Quarteroni$^+$ 09] is software and hardware repairing in the area of an IT help-desk. It consists of human-machine dialogs acquired with a Wizard of Oz approach. The data, containing approximately 40 different concepts, are split into training, development and test sets made of 3k, 400 and 640 sentences respectively. An example sentence from the Italian corpus including the attribute name and value tags is given in Figure A.5.

The semantic annotation is context dependent at turn level, meaning that the same words can be associated with different concepts depending on the object they refer to (for example `it is not working` can be **SoftwareProblem** or **HardwareProblem**). This, together with the very spontaneous form of user turns, makes the task rather complex despite the relatively small number of concepts to be distinguished.

A.2 English NETtalk 15k

The original NETtalk corpus contains the phonetic transcription of 20,008 English words as well as stress and syllabic structure information per word. This corpus has been created to train a backpropagation network for the G2P task and has been first proposed in [Sejnowski & Rosenberg 86, Sejnowski & Rosenberg 87]. Since an alignment between graphemes and phonemes was needed where the grapheme side is always longer than the phoneme side, the NETtalk

dzień dobry ja chciałabym poprosić
o trasę autobusu sto dwadzieścia siedem

Good day I would like to ask about
the bus route one hundred twenty-seven

(a) Manual transcription with English translation for reference (not included in the corpus).

CONV_FORM{dzień dobry} **NULL**{ja} **Action**{chciałabym}
Action{poprosić} **NULL**{o} **ROUTE**{trasę}
BUS{autobusu sto dwadzieścia siedem}

(b) Attribute name tags for the example sentence.

CONV_FORM[HELLO] **Action**[Request]
Action[Request] **ROUTE**[General] **BUS**[sto dwadzieścia siedem]

(c) Final attribute name and attribute value sequence for the example sentence.

Figure A.4 Example sentence from the Polish SLU corpus with attribute name and value concept tags. Some attribute values are provided in Polish like e.g. numbers.

corpus has been constructed in such a way, e.g. for the grapheme "x" several new phonemes, namely /X/, /K/, and /#/, have been introduced to avoid the double phonemes /ks/, /kS/, and /gz/ and thus a one-to-two mapping. Also, all words have been converted to lower-case. The corpus is freely available for research purposes as part of the Letter-to-Phoneme Conversion Challenge by Pattern Analysis, Statistical Modelling and Computational Learning (PASCAL), a European Commission's IST-funded Network of Excellence for Multimodal Interfaces [van den Bosch & Chen$^+$ 06] as well as on the website of Cambridge University [Sejnowski & Rosenberg 93].

There is also a version with a manually annotated one-to-one alignment available, which has been used for the reported experiments. Due to the nature of the corpus design, only a "NULL" or empty phoneme had to be introduced, since the phoneme side is by design at most as long as the grapheme side. Although introducing only one empty phoneme might be linguistically correct, there is a major drawback for statistical modelling: since this empty phoneme can more or less occur anywhere in a word, there is no context information to reliably learn where to hypothesize such a symbol. To overcome this drawback, we introduced so-called named epsilons, as depicted in Figure A.6. In Figure A.6 (a), the original one-to-one alignment is shown. The first step now is to prepend all εs with the last non-empty phoneme (cf. Figure A.6 (b)). The resulting new phonemes are called named epsilons. By utilizing the BIO scheme, this alignment can be transformed into a nearly ε-free one-to-one alignment as shown in Figure A.6 (c). Here, all original epsilons get a "start" marker, whereas the following

Buongiorno io ho un problema con la stampante
da questa mattina non riesco piu' a stampare

Good morning I have a problem with the printer
since this morning I cannot print any more

(a) Manual transcription with English translation for reference (not included in the corpus).

null{Buongiorno io ho} **HardwareProblem.type**{un problema}
Peripheral.type{con la stampante} **Time.relative**{da questa mattina}
HardwareOperation.negate{non riesco} **null**{piu'}
HardwareOperation.operationType{a stampare}

(b) Attribute name tags for this example including modes.

HardwareProblem.type[general_problem] **Peripheral.type**[printer]
Time.relative[morning] **HardwareOperation.negate**[non]
HardwareOperation.operationType[to_print]

(c) Final attribute name and attribute value sequence for this example.

Figure A.5 Example sentence from the Italian corpus with attribute name and value concept tags.

Table A.2 Statistics of the English NETtalk pronunciation dictionary.

	# symbols		∅ word length		∅ prons/	# words		
	source	target	source	target	word	train	dev	eva
NETTalk 15k	26	50	7.3	6.2	1.010	13,804	1,071	4,951

named epsilons get a "continue" marker. It is easily possible to derive a many-to-one alignment by aggregating the "start" and "continue" phonemes as shown in Figure A.6 (d).

Note that we still retain the empty phoneme as is for words starting with letters which are not pronounced; cf. Figure A.7 for an example.

Albeit the somewhat artificial alignment constraint, the corpus is well suited to test and develop G2P methods due to its comparatively small size and the one-to-one alignment between graphemes and phonemes rendering a complex alignment algorithm unnecessary.

In the reported experiments in this work, we use the so-called NETtalk 15k split, which roughly leaves 5k words for evaluation. This split has been suggested in [Jiang & Hon$^+$ 97] and is frequently used in the literature. A small part from the 15k training words has been set apart to form a development set. The complete data statistics are presented in Table A.2.

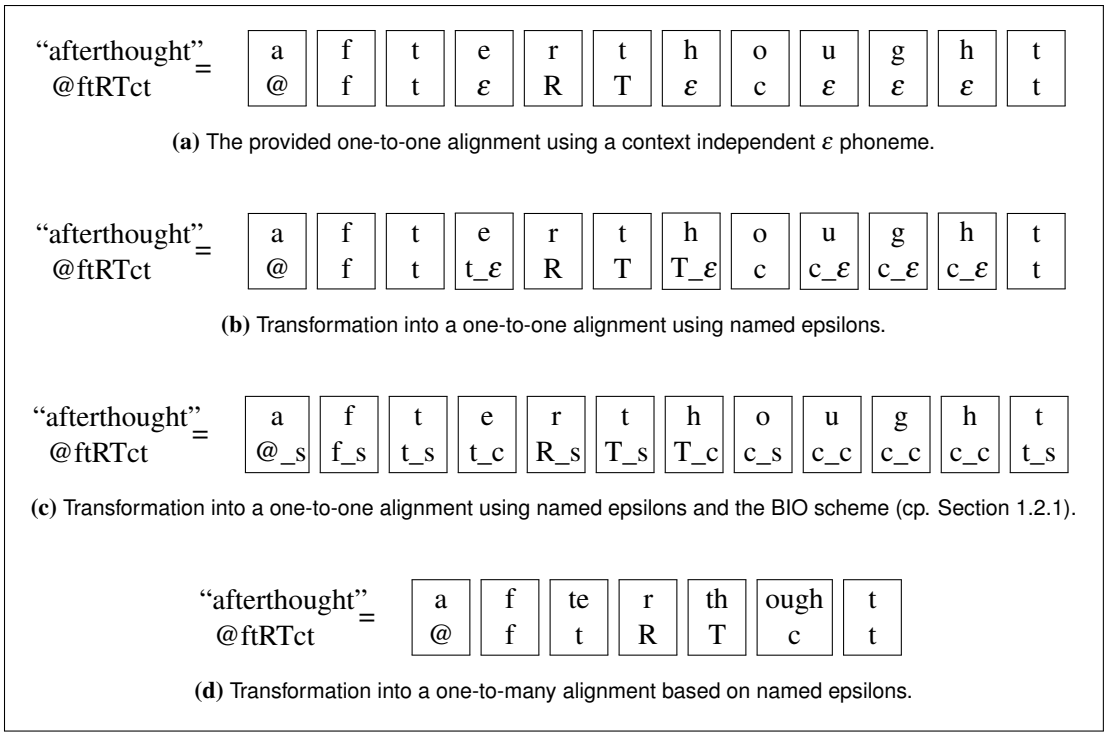


Figure A.6 Example for modeling one-to-many alignments using so-called named epsilons and the BIO scheme from the NETtalk 15k corpus. ε denotes the empty phoneme.

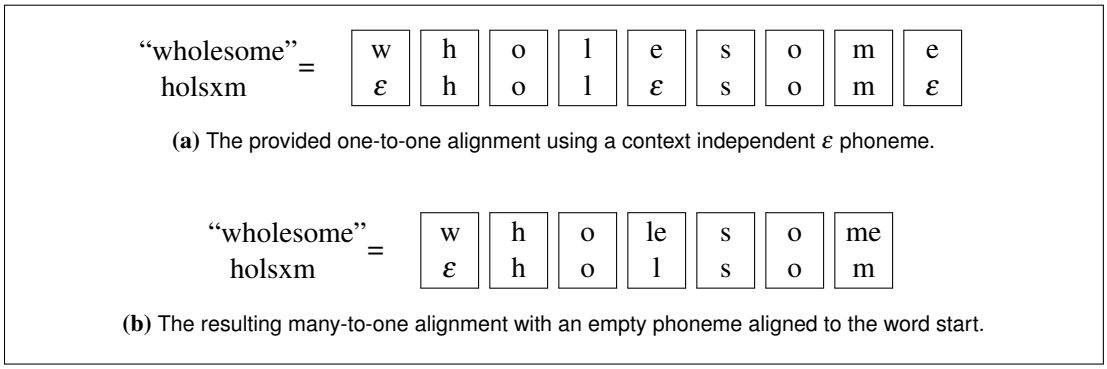


Figure A.7 Example of a many-to-one alignment from the NETtalk 15k corpus with the empty phoneme, denoted as ε.

Table A.3 Statistics of the English CELEX pronunciation dictionary.

# symbols		∅ word length		# prons/	# unique words		
source	target	source	target	word	train	dev	eva
26	58	8.4	7.1	1.000	39,985	5,000	15,000

A.3 English CELEX

The English CELEX corpus is a randomly selected subset of the original English CELEX database [Baayen & Piepenbrock+ 96]. For this subset, phrases, abbreviations and homographs have been discarded. Additionally, all words have been converted to lower case. It consists of 60,000 pronunciations, whereas only 15 words have more than one pronunciation variant, namely two. Thus, there are 59,985 different lemmata covered in this dictionary. Since CELEX is based upon various British and American text sources, it contains a mixture of both pronunciation styles, whereas at most 15.4% of the data is of American English origin. The available data has been split into three sets: 40k lemmata for training, 5k for development and 15k for testing. The 15 words with pronunciation variants have been put into the training set. Thus, all entries in the development and testing corpora have exactly one reference pronunciation. In total, 26 different graphemes representing all lower case letters of the English alphabet and 58 phonemes are used. The phonemes are represented using SAMPA notation, which has been especially designed to be computer-readable and uses only 7-bit printable American Standard Code for Information Interchange (ASCII) characters. Within the original CELEX corpus, diphthongs and monophthongs are annotated with character sequences greater than one, e.g. the diphthong "ai" in the English word stairs is transcribed as "e@". The disadvantage in using a character sequence instead of a single symbol is that it is not easily possible to automatically segment a given pronunciation into phonemes. Thus, for the CELEX corpus used in this work, all phonemes describing diphthongs and monophthongs have been mapped to single characters. E.g., the aforementioned diphthong "ai" is represented as "8". If the original SAMPA annotation is needed, the mapping is easily reversible since it is bijective. The complete corpus statistics are given in Table A.3.

Despite the lack of pronunciation variants, this corpus is often used in publications dealing with small monotone translation tasks like G2P and thus a good choice for comparison with other groups. For example, results on this corpus are reported in [Bisani & Ney 02, Chen 03, Vozila & Adams+ 03, Bisani & Ney 08, Jiampojamarn & Cherry+ 10, Lehnen & Hahn+ 11a, Hahn & Vozila+ 12]. In this work, experimental results using the CELEX corpus are reported in Section 6.2.2.

A.4 Western-European LVCSR Dictionaries and Corpora

To get an idea of the performance of G2P systems on real state-of-the-art LVCSR tasks, some study sets have been built in various languages. This comprises building a pronunciation dictio-

Table A.4 Data statistics for pronunciation dictionaries in various Western-European languages (study sets).

language	# symbols		∅ word length		# unique
	source	target	source	target	words
English	76	50	8.8	7.6	283k
German	65	48	13.3	12.1	436k
French	72	38	10.3	7.7	319k
Italian	63	38	10.2	10.1	252k
Dutch	66	45	11.5	9.9	347k

nary for G2P model training and testing, building an ASR system, as well as defining ASR evaluation sets per language. We have chosen five Western-European languages, namely (British) English, German, French, Italian, and Dutch.

The statistics for the pronunciation dictionaries are given in Table A.4. The average number of pronunciations per word is < 1.09 for all lexica. We randomly selected 5% of the data for the development and evaluation set each and kept these sets fixed for all experiments. Using these sets, the G2P systems have been built and evaluated. Note that the number of source symbols denotes symbols seen at least ten times in the training data, including upper and lower case letters, accented characters as well as punctuation marks. In total, 151 characters are included at least once per language. Since there are compounds included in the German data set, the average word length as well as the average number of phonemes per word is higher than for any of the other selected languages.

Concerning the LVCSR evaluation sets, the corresponding statistics are given in Table A.5. For all considered languages except Dutch, the audio data comprises more than 150 hours of speech. The vocabulary varies between 20k and 181k words. Since we want to evaluate our methods on real-life tasks, the G2P ratio, i.e. the percentage of pronunciations within the pronunciation dictionary which have been generated using a G2P system, is rather small (ranging from 0.31% to 1.66%). All other pronunciations are taken from a manually designed dictionary and/or have been verified by a linguist. The OOV rates have been calculated w.r.t. the recognition lexicons used within the ASR experiments.

A.5 English QUAERO Corpora

Within the French-German QUAERO project[2], one research focus is on improving LVCSR for various European languages, especially on more difficult data like web podcasts or mixed shows including e.g. on the street interviews with background noise, overlapping speech, mixed acoustic conditions, fast and colloquial speaking styles, etc. To facilitate ASR model training for these conditions, data matching these conditions has been continuously collected as well as manually transcribed for training and testing purposes. In each year of the project, an evaluation

[2] http://www.quaero.org

Table A.5 Data statistics for the various LVCSR study sets. G2P ratio denotes the portion of the evaluation set for which the pronunciation(s) in the corresponding recognition lexicon have been generated with a G2P model.

	total data[h]	running words	vocab size	G2P ratio [%]	OOV ratio [%]
English	157.3	629k	39k	0.44	1.00
German	184.5	719k	81k	0.80	3.22
French	177.7	807k	126k	1.66	1.81
Italian	176.7	627k	54k	0.31	2.51
Dutch	31.3	231k	20k	1.24	2.35

Table A.6 Statistics of the used QUAERO English corpora.

data set	duration[h]	# running words
train11	234.3	2.8M
dev10	3.3	40K
eval10	3.8	45K
eval11	3.3	35K
eval12	3.4	40K

takes place on a previously unseen test set, whereas usually the test sets from former evaluations may be used for development.

For the reported experiments, we have chosen a state-of-the-art English LVCSR task based on the QUAERO 2011 data [Sundermeyer & Nußbaum-Thom$^+$ 11]. The data for training comprises roughly 234h of audio data and mainly consists of broadcast news and podcasts. There are four datasets for development and evaluation of ASR systems provided which have a duration between three and four hours and are comprised of 35K to 45K running words. An overview of the used ASR data is given in Table A.6.

For the training of the acoustic model as well as the G2P models, we have chosen the BEEP dictionary [Robinson 95] as a background lexicon, comprising roughly 257k pronunciations for 238k words, which have been partly derived from the Oxford Text Archive releases 710 and 1054. The original BEEP lexicon is delivered in all upper case, which has been converted to all lower case. Concerning the training of the G2P model, besides the 26 letters of the English alphabet, the single quote (') which is mostly used for the genitive 's as well as the hyphen is kept, whereas all words containing any other character are removed. Additionally, especially marked compound words as well as abbreviations are also removed, since they usually do not help in learning letter-to-sound dependencies. The statistics for the resulting pronunciation dictionary comprising 237k entries are presented in Table A.7.

For the training of G2P models, we split the data into a training and a development set, whereas the latter contains roughly 10K words (4% of the total data). The split has been done

Table A.7 Statistics of the BEEP pronunciation dictionary which has been used as background lexicon for the QUAERO English ASR system and thus also as training data for the respective G2P models.

# symbols		∅ word length		∅ prons	# unique
source	target	source	target	per word	words
28	44	9.03	7.60	1.08	237k

randomly and all pronunciation variants of a certain word are either in the training set or in the development set, but never spread across both. To avoid any encoding issues, all the data has been converted to UTF-8 in a preprocessing step.

Compared to e.g. the CELEX dictionary as presented in Section A.3, the number of phonemes is rather small. This is typical for pronunciation dictionaries used within ASR tasks and has the following reason. Since for the training of the acoustic model there have to be a certain minimal amount of training samples available for each phoneme (more precisely: for as many as possible triphones), phonemes which do rarely occur are mapped to the linguistically closest phoneme. This procedure is a kind of bottom-up clustering and necessary to ensure that the model is robust.

List of Figures

List of Tables

Glossary

allophone
: A single phoneme might have various slightly different pronunciations depending on the phoneme context. Such an actual realization of a phoneme is referred to as allophone.

anaphora
: A linguistic term describing the link of a constituent to a preceding constituent, e.g. in the sentence “Peter did invite Paul, but he could not make it.”, “he” is anaphoric and refers to Paul.

attribute name
: A tag representing the semantic meaning of a word sequence or chunk. The attribute name is required for each concept.

attribute value
: A normalized value which may be associated with an attribute name. It also depends on the corresponding word sequence and has to be extracted additionally.

compound
: If a new word is built by concatenating two or more existing words, the resulting word is called a compound. For example, the German compound word “Kindergarten” consists of the two nouns “Kinder” (children) and “Garten” (garden).

concept
: A set of attributes which is assigned to a sequence of words. This set contains up to two elements: the attribute name and the attribute value.

diphthong
: Within a long vowel, two vowel sounds occur one after another in the same syllable. Examples in English would be the “ai” sound in “knives” or the “ou” sound at the end of “snow”.

homograph
: A group of words which are spelled the same but which have different meaning and/or pronunciation, e.g. the English word “content”, which could either be an adjective (with stress on the second syllable) or a noun (with stress on the first syllable).

monophthong

A pure vowel sound in which the articulation does not change from the beginning of the vowel to the end. An example in English would be the double e in “teeth” or the first two vowels in “ease”.

UTF-8

Abbreviation for 8-Bit Universal Character Set (UCS) Transformation Format. A widely spread encoding which can represent any character within the Unicode character set, which comprises more or less all written characters of all languages. It is frequently used in text processing, since it avoids shifting between encodings for different languages.

Acronyms

CELEX	The Dutch Centre for Lexical Information
AM	acoustic model
ARPA	Advanced Research Projects Agency
ASCII	American Standard Code for Information Interchange
ASR	automatic speech recognition
ATIS	Air Travel Information System
BEEP	British English Example Pronunciation
BFGS	Broyden-Fletcher-Goldfarb-Shanno
BIO	begin insides outside scheme
C-MLLR	constrained MLLR
CART	classification and regression tree
CER	concept error rate
CoNLL	Conference on Computational Natural Language Learning
CPU	central processing unit
CRF	conditional random field
CSI	Consorzio per il Sistema Informativo Piemonte
DBN	dynamic Bayesian network
DNA	desoxyribonucleic acid
EM	expectation maximization
EN	elastic net
FLM	factored language model
FSA	finite state automaton
FST	finite state transducer
G2P	grapheme-to-phoneme conversion
GMM	Gaussian mixture model
GPB	generalized parallel backoff
GT	gammatone filter based features

HCRF	hidden conditional random field
HDCRF	hidden dynamic conditional random field
HMM	hidden Markov model
HMRASTA	hierarchical multiple RASTA
IBM	International Business Machines Corporation
IP	Integrated Project
IRISA	Institut de Recherche en Informatique et Systèmes Aléatoires
IST	Information Society Technologies
LBFGS	limited memory BFGS
LDA	linear discriminant analysis
LM	language model
LUNA	Spoken Languange UNderstanding in MultilinguAl Communication Systems
LVCSR	large vocabulary continuous speech recognition
M-MMI	margin-based MMI
MAP	maximum a posteriori
ME	maximum entropy
MEMM	maximum entropy Markov model
MERT	minimum error rate training
MFCC	mel-frequency cepstral coefficients
MIT	Massachusetts Institute of Technology
ML	maximum likelihood
MLLR	maximum likelihood linear regression
MLP	multi-layer perceptron
MMI	maximum mutual information
MPE	minimum phone error
MRL	meaning representation language
MT	machine translation
NIST	National Institute for Standards and Technology
NLU	natural language understanding
NN	neural network
NP	noun phrase
OOV	out of vocabulary
OWL-QN	orthant-wise quasi-Newton

PASCAL	Pattern Analysis, Statistical Modelling and Computational Learning
PBT	phrase-based machine translation
PER	phoneme error rate
PLP	perceptual linear prediction
POS	part-of-speech
PTK	partial tree kernel
RASTA	relative spectra
RNN	recurrent neural network
ROVER	recognizer output voting error reduction
RProp	resilient backpropagation
RWTH	RWTH Aachen University
RWTH ASR	RWTH Aachen University Speech Recognition
SAMPA	Speech Assessment Methods Phonetic Alphabet
SER	sentence error rate
SGD	stochastic gradient descent
SIMD	single instruction multiple data
SLU	spoken language understanding
SMT	statistical machine translation
SRI	SRI International
SVM	support vector machine
VTLN	vocal tract length normalization
WER	word error rate
YamCha	Yet Another Multipurpose CHunk Annotator

List of Symbols

ε	denotes the empty token (FST) or the empty symbol
<s>	denotes the sentence start symbol
</s>	denotes the sentence end symbol
π	an FST operation denoting the removal of unwated symbols like ε, <s>, </s>

Bibliography

[Acero 90] A. Acero: *Acoustical and Environmental Robustness in Automatic Speech Recognition*. Ph.D. thesis, Carnegie Mellon University, Pittsburgh, PA, USA, Sept. 1990.

[Aertsen & Johannesma$^+$ 80] A.M.H.J. Aertsen, P.I.M. Johannesma, D.J. Hermes: Spectro-Temporal Receptive Fields of Auditory Neurons in the Grassfrog. *Biological Cybernetics*, Vol. 38, pp. 235 – 248, Nov. 1980.

[Allauzen & Riley$^+$ 07] C. Allauzen, M. Riley, J. Schalkwyk, W. Skut, M. Mohri: OpenFst: a General and Efficient Weighted Finite-State Transducer Library. In *Proc. Int. Conf. on Implementation and Application of Automata*, pp. 11 – 23, Prague, Czech Republic, July 2007.

[Alleva & Huang$^+$ 96] P. Alleva, X.D. Huang, M.Y. Hwang: Improvements on the Pronunciation Prefix Tree Search Organization. In *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, Vol. 1, pp. 133 – 136, Atlanta, GA, USA, May 1996.

[Arnold & Balkan$^+$ 94] D. Arnold, L. Balkan, L.L. Humphreys, S. Meijer, L. Sadler: *Machine Translation*. Blackwell Publishers, 1994.

[Baayen & Piepenbrock$^+$ 96] R. Baayen, R. Piepenbrock, L. Gulikers: CELEX2, 1996.

[Bahl & Jelinek$^+$ 83] L.R. Bahl, F. Jelinek, R.L. Mercer: A Maximum Likelihood Approach to Continuous Speech Recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 5, pp. 179 – 190, March 1983.

[Baker 75] J.K. Baker: Stochastic Modeling for Automatic Speech Understanding. In D.R. Reddy, editor, *Speech Recognition*, pp. 512 – 542. Academic Press, New York, NY, USA, 1975.

[Bakis 76] R. Bakis: Continuous Speech Word Recognition via Centisecond Acoustic States. *Journal of the Acoustical Society of America*, Vol. 59, No. 1, pp. 97, April 1976.

[Bar-Hillel 51] Y. Bar-Hillel: The Present State of Research on Mechanical Translation. *American Documentation*, Vol. 2, pp. 229 – 237, 1951.

[Basha Shaik & El-Desoky Mousa$^+$ 11] M.A. Basha Shaik, A. El-Desoky Mousa, R. Schlüter, H. Ney: Hybrid Language Models Using Mixed Types of Sub-lexical Units for Open Vocabulary German LVCSR. In *Proc. Int. Conf. on Speech Communication and Technology (Interspeech)*, pp. 1441 – 1444, Florence, Italy, Aug. 2011.

[Baum 72] L.E. Baum: An Inequality and Associated Maximization Technique in Statistical Estimation for Probabilistic Functions of Markov Processes. In O. Shisha, editor, *Inequalities*, Vol. 3, pp. 1 – 8. Academic Press, New York, NY, USA, 1972.

[Bayes 63] T. Bayes: An Essay towards solving a problem in the doctrine of chances. *Philosophical Transactions of the Royal Society of London*, Vol. 53, pp. 370 – 418, Jan. 1763. Reprinted in *Biometrika*, vol. 45, no. 3/4, pp. 293–315, December 1958.

[Bellman 57] R.E. Bellman: Dynamic Programming. Princeton University Press, Princeton, NJ, USA, 1957.

[Bender & Macherey$^+$ 03] O. Bender, K. Macherey, F.J. Och, H. Ney: Comparison of Alignment Templates and Maximum Entropy Models for Natural Language Understanding. In *Proc. Conf. of the European Chapter of the Association for Computational Linguistics*, pp. 11 – 18, Budapest, Hungary, April 2003.

[Berger & Brown$^+$ 96] A.L. Berger, P.F. Brown, S.A.D. Pietra, V.J.D. Pietra, J.R. Gillett, A.S.. Kehler, R.L. Mercer: Language Translation Apparatus and Method of Using Context-based Translation Models, United States Patent, Patent Number 5510981, April 1996.

[Beulen 99] K. Beulen: *Phonetische Entscheidungsbäume für die automatische Spracherkennung mit großem Vokabular*. Ph.D. thesis, RWTH Aachen University, Aachen, Germany, July 1999.

[Bilmes & Kirchhoff 03] J.A. Bilmes, K. Kirchhoff: Factored language models and generalized parallel backoff. In *Proc. Conf. of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pp. 4 – 6, Edmonton, Canada, May 2003. Association for Computational Linguistics.

[Birch & Blunsom$^+$ 09] A. Birch, P. Blunsom, M. Osborne: A Quantitative Analysis of Reordering Phenomena. In *Proc. Workshop on Statistical Machine Translation*, pp. 197 – 205, Athens, Greece, March 2009. Association for Computational Linguistics.

[Bisani 08] M. Bisani: Sequituri G2P. http://www-i6.informatik.rwth-aachen.de/web/Software/g2p.html, 2008.

[Bisani & Ney 02] M. Bisani, H. Ney: Investigations on Joint-Multigram Models for Grapheme-to-Phoneme Conversion. In *Proc. Int. Conf. on Spoken Language Processing*, pp. 105–108, Denver, CO, USA, Sept. 2002.

[Bisani & Ney 05] M. Bisani, H. Ney: Open Vocabulary Speech Recognition with Flat Hybrid Models. In *Proc. Int. Conf. on Speech Communication and Technology (Interspeech)*, pp. 725 – 728, Sept. 2005.

[Bisani & Ney 08] M. Bisani, H. Ney: Joint-Sequence Models for Grapheme-to-Phoneme Conversion. *Speech Communication*, Vol. 50, No. 5, pp. 434 – 451, May 2008.

[Bo-June & Glass 08] P.H. Bo-June, J. Glass: Iterative Language Model Estimation: Efficient Data Structure & Algorithms. In *Proc. Int. Conf. on Speech Communication and Technology (Interspeech)*, pp. 841 – 844, Brisbane, Australia, Sept. 2008.

[Bonneau-Maynard & Ayache$^+$ 06] H. Bonneau-Maynard, C. Ayache, F. Béchet, A. Denis, A. Kuhn, F. Lefèvre, D. Mostefa, M. Qugnard, S. Rosset, J. Servan, S. Vilaneau: Results of the French Evalda-Media Evaluation Campaign for Literal Understanding. In *Proc. Int. Conf. on Language Resources and Evaluation*, pp. 2054 – 2059, Genoa, Italy, May 2006.

[Bonneau-Maynard & Rosset$^+$ 05] H. Bonneau-Maynard, S. Rosset, C. Ayache, A. Kuhn, D. Mostefa: Semantic annotation of the French MEDIA dialog corpus. In *Proc. European Conf. on Speech Communication and Technology*, pp. 3457 – 3460, Lisboa, Portugal, Sept. 2005.

[Brown & Cocke$^+$ 88] P.F. Brown, J. Cocke, S.A. Della Pietra, V.J. Della Pietra, F. Jelinek, J.D. Lafferty, R.L. Mercer, P.S. Rossin: A Statistical Approach to Language Translation. In *Proceedings of the 12th Conference on Computational Linguistics*, pp. 71 – 76, Buffalo, NY, USA, Aug. 1988.

[Brown & Cocke$^+$ 90] P.F. Brown, J. Cocke, S.A. Della Pietra, V.J. Della Pietra, F. Jelinek, J.D. Lafferty, R.L. Mercer, P.S. Rossin: A Statistical Approach to Machine Translation. *ACL Computational Linguistics*, Vol. 16, No. 2, pp. 79–85, June 1990.

[Brown & Della Pietra$^+$ 93] P.F. Brown, S.A. Della Pietra, D.V. J., R.L. Mercer: The Mathematics of Statistical Machine Translation: Parameter Estimation. *ACL Computational Linguistics*, Vol. 19, No. 2, pp. 263 – 311, 1993.

[Brown & Pietra$^+$ 92] P.F. Brown, V.J.D. Pietra, R.L. Mercer, S.A.D. Pietra, J.C. Lai: An Estimate of an Upper Bound for the Entropy of English. *ACL Computational Linguistics*, Vol. 18, No. 1, pp. 31 – 40, March 1992.

[Camelin & Béchet$^+$ 10] N. Camelin, F. Béchet, G. Damnati, R. De Mori: Detection and Interpretation of Opinion Expressions in Spoken Surveys. *IEEE Trans. on Audio, Speech and Language Processing*, Vol. 18, No. 2, pp. 369–381, 2010.

[Caseiro & Trancoso$^+$ 02] D. Caseiro, I. Trancoso, L. Oliveira, C. Viana: Grapheme-to-Phone using Finite-State Transducers. In *IEEE Workshop on Speech Synthesis*, pp. 215 – 218, Santa Monica, CA, USA, Sept. 2002.

[Chen 03] S.F. Chen: Conditional and Joint Models for Grapheme-to-Phoneme Conversion. In *Proc. European Conf. on Speech Communication and Technology*, pp. 2033 – 2036, Geneva, Switzerland, Sept. 2003.

[Chen & Goodman 96] S.F. Chen, J. Goodman: An Empirical Study of Smoothing Techniques for Language Modeling. In *Proc. Annual Meeting of the Association for Computational Linguistics*, pp. 310 – 318, Santa Cruz, CA, USA, June 1996.

[Chen & Goodman 98] S.F. Chen, J. Goodman: An Empirical Study of Smoothing Techniques for Language Modeling. Technical Report 10, Center for Research in Computing Technology (Harvard University), pp. 1 – 63, Aug. 1998.

[Chen & Rosenfeld 00] S. Chen, R. Rosenfeld: A Survey of Smoothing Techniques for ME Models. *IEEE Transactions on Speech and Audio Processing*, Vol. 8, No. 1, pp. 37 – 50, Jan. 2000.

[Chiang 05] D. Chiang: A Hierarchical Phrase-Based Model for Statistical Machine Translation. In *Proc. Annual Meeting of the Association for Computational Linguistics*, pp. 263 – 270, Ann Arbor, Michigan, MI, USA, June 2005.

[Chiang 07] D. Chiang: Hierarchical Phrase-Based Translation. *ACL Computational Linguistics*, Vol. 33, No. 2, pp. 201 – 228, June 2007.

[Collins & Duffy 02] M. Collins, N. Duffy: New Ranking Algorithms for Parsing and Tagging: Kernels over Discrete Structures, and the Voted Perceptron. In *Proc. Annual Meeting of the Association for Computational Linguistics*, pp. 263–270, Philadelphia, PA, USA, July 2002.

[CoNLL-2000 00] CoNLL-2000: Results of the CoNLL-2000 Shared Task on Chunking, 2000. http://www.cnts.ua.ac.be/conll2000/chunking/.

[Cortes & Vapnik 95] C. Cortes, V. Vapnik: Support-Vector Networks. *Machine Learning*, Vol. 20, pp. 273 – 297, 1995.

[Damnati & Béchet$^+$ 07] G. Damnati, F. Béchet, R. de Mori: Spoken Language Understanding Strategies on the France Telecom 3000 Voice Agency Corpus. In *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, Vol. 4, pp. 9 – 12, Honolulu, HI, USA, April 2007.

[Davis & Biddulph$^+$ 51] K. Davis, R. Biddulph, S. Balashek: Automatic Recognition of Spoken Digits. *The Journal of the Acoustic Society of America*, Vol. 24, No. 6, pp. 637 – 642, 1951.

[Davis & Mermelstein 80] S. Davis, P. Mermelstein: Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences. *IEEE Trans. on Acoustics, Speech, and Signal Processing*, Vol. 28, No. 4, pp. 357 – 366, Aug. 1980.

[De Mori & Hakkani-Tur$^+$ 08] R. De Mori, D. Hakkani-Tur, M. McTear, G. Riccardi, G. Tur: Spoken Language Understanding: a Survey. *IEEE Signal Processing Magazine*, Vol. 25, pp. 50–58, 2008.

[Deligne & Yvon$^+$ 95] S. Deligne, F. Yvon, F. Bimbot: Variable-Length Sequence Matching for Phonetic Transcription using Joint Multigrams. In *Proc. European Conf. on Speech Communication and Technology*, pp. 2243 – 2246, Madrid, Spain, Sept. 1995.

[Dempster & Laird$^+$ 77] A. Dempster, N. Laird, D. Rubin: Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society*, Vol. 39, No. B, pp. 1 – 38, 1977.

[Deoras & Sarikaya$^+$ 12] A. Deoras, R. Sarikaya, G. Tur, D. Hakkani-Tur: Joint Decoding for Speech Recognition and Semantic Tagging. In *Proc. Int. Conf. on Speech Communication and Technology (Interspeech)*, Portland, OR, USA, Sept. 2012. Electronic, no page numbers.

[Deselaers & Hasan$^+$ 09] T. Deselaers, S. Hasan, O. Bender, H. Ney: A Deep Learning Approach to Machine Transliteration. In *Proc. of the EACL Int. Workshop on Statistical Machine Translation*, pp. 233 – 241, Athens, Greece, March 2009.

[Dijkstra 59] E.W. Dijkstra: A Note on Two Problems in Connection with Graphs. *Numerische Mathematik*, Vol. 1, pp. 269 – 271, 1959.

[Dinarelli 10] M. Dinarelli: *Spoken Language Understanding: from Spoken Utterances to Semantic Structures*. Ph.D. thesis, International Doctoral School in Information and Communication Technology, Dipartimento di Ingegneria e Scienza dell' Informazione, via Sommarive 14, 38100 Povo di Trento (TN), Italy, March 2010.

[Dinarelli & Moschitti$^+$ 09a] M. Dinarelli, A. Moschitti, G. Riccardi: Re-ranking Models Based on Small Training Data for Spoken Language Understanding. In *Proc. Conf. of Empirical Methods for Natural Language Processing*, pp. 11 – 18, Singapore,Singapore, Aug. 2009.

[Dinarelli & Moschitti$^+$ 09b] M. Dinarelli, A. Moschitti, G. Riccardi: Re-ranking Models for Spoken Language Understanding. In *Proc. Conf. of the European Chapter of the Association for Computational Linguistics*, pp. 202 – 210, Athens,Greece, April 2009.

[Dinarelli & Moschitti$^+$ 12] M. Dinarelli, A. Moschitti, G. Riccardi: Discriminative Reranking for Spoken Language Understanding. *IEEE Trans. on Audio, Speech and Language Processing*, Vol. 20, pp. 526 – 539, 2012.

[Dinarelli & Quarteroni$^+$ 09] M. Dinarelli, S. Quarteroni, S. Tonelli, A. Moschitti, G. Riccardi: Annotating Spoken Dialogs: from Speech Segments to Dialog Acts and Frame Semantics. In *Proc. of the EACL Int. Workshop on Semantic Representation of Spoken Language*, SRSL '09, pp. 34 – 41, Stroudsburg, PA, USA, March 2009. Association for Computational Linguistics.

[Doddington & Przybocki$^+$ 00] G.R. Doddington, M.A. Przybocki, A.F. Martin, D.A. Reynolds: The NIST Speaker Recognition Evaluation – Overview, Methodology, Systems, Results, Perspective. *Speech Communication*, Vol. 31, No. 2 – 3, pp. 225 – 254, June 2000.

[Duda & Hart$^+$ 01] R.O. Duda, P.E. Hart, D.G. Stork: *Pattern Classification*. John Wiley & Sons, New York, NY, USA, 2001.

[Duvert & Meurs+ 08] F. Duvert, M.J. Meurs, C. Servan, F. Bechét, F. Lefèvre, R. De Mori: Semantic Composition Process in a Speech Understanding System . In *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, pp. 5029 – 5032, Las Vegas, NV, USA, March 2008.

[El-Desoky Mousa & Basha Shaik+ 12] A. El-Desoky Mousa, M.A. Basha Shaik, R. Schlüter, H. Ney: Morpheme Level Feature-based Language Models for German LVCSR. In *Proc. Int. Conf. on Speech Communication and Technology (Interspeech)*, Portland, OR, USA, Sept. 2012. Electronic, no page numbers.

[Fiscus 97] J.G. Fiscus: A Post-Processing System to Yield Reduced Word Error Rates: Recogniser Output Voting Error Reduction (ROVER). In *Proc. IEEE Automatic Speech Recognition and Understanding Workshop*, pp. 347 – 352, Santa Barbara, CA, USA, Dec. 1997.

[Fisher 36] R.A. Fisher: The Use of Multiple Measurements in Taxonomic Problems. *Annals of Eugenics*, Vol. 7, No. 179 – 188, 1936.

[Fritsch 97] J. Fritsch: ACID/HNN: A Framework for Hierarchical Connectionist Acoustic Modeling. In S. Furui, B.H. Juang, W. Chou, editors, *Proc. IEEE Automatic Speech Recognition and Understanding Workshop*, pp. 164 – 171, Santa Barbara, CA, USA, Dec. 1997.

[Galescu & Allen 02] L. Galescu, J.F. Allen: Bi-directional Conversion Between Graphemes and Phonemes Using a Joint N-gram Model. In *ISCA Tutorial and Research Workshop (ITRW) on Speech Synthesis*, Rochester, NY, USA, Aug. 2002. paper 131; no page numbers.

[Generet & Ney+ 95] M. Generet, H. Ney, F. Wessel: Extensions to Absolute Discounting for Language Modeling. In *Proc. European Conf. on Speech Communication and Technology*, Vol. 2, pp. 1245 – 1248, Madrid, Spain, Sept. 1995.

[Gollan & Ney 08] C. Gollan, H. Ney: Towards Automatic Learning in LVCSR: Rapid Development of a Persian Broadcast Transcription System. In *Proc. Int. Conf. on Speech Communication and Technology (Interspeech)*, pp. 1441 – 1444, Brisbane, Australia, Sept. 2008.

[Guta 11] A. Guta: Grapheme to Phoneme Conversion Using CRFs with Integrated Alignments. Diploma thesis, RWTH Aachen University, May 2011.

[Hahn & Dinarelli+ 11] S. Hahn, M. Dinarelli, C. Raymond, F. Lefèvre, P. Lehnen, R. De Mori, A. Moschitti, H. Ney, G. Riccardi: Comparing Stochastic Approaches to Spoken Language Understanding in Multiple Languages. *IEEE Trans. on Audio, Speech and Language Processing*, Vol. 19, No. 6, pp. 1569 – 1583, Aug. 2011.

[Hahn & Lehnen+ 08a] S. Hahn, P. Lehnen, H. Ney: System Combination for Spoken Language Understanding. In *Proc. Int. Conf. on Speech Communication and Technology (Interspeech)*, pp. 236 – 239, Brisbane, Australia, Sept. 2008.

[Hahn & Lehnen+ 08b] S. Hahn, P. Lehnen, C. Raymond, H. Ney: A Comparison of Various Methods for Concept Tagging for Spoken Language Understanding. In *Proc. Int. Conf. on*

Language Resources and Evaluation, Marrakech, Morocco, May 2008. Electronic, no page numbers.

[Hahn & Lehnen$^+$ 09] S. Hahn, P. Lehnen, G. Heigold, H. Ney: Optimizing CRFs for SLU Tasks in Various Languages Using Modified Training Criteria. In *Proc. Int. Conf. on Speech Communication and Technology (Interspeech)*, pp. 2727 – 2730, Brighton, UK, Sept. 2009.

[Hahn & Lehnen$^+$ 11] S. Hahn, P. Lehnen, H. Ney: Powerful Extensions to CRFs for Grapheme to Phoneme Conversion. In *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, pp. 4912 – 4915, Prague, Czech Republic, May 2011.

[Hahn & Lehnen$^+$ 13] S. Hahn, P. Lehnen, S. Wiesler, R. Schlüter, H. Ney: Improving LVCSR with Hidden Conditional Random Fields for Grapheme-to-Phoneme Conversion. In *Proc. Int. Conf. on Speech Communication and Technology (Interspeech)*, Lyon, France, Aug. 2013. Electronic, no page numbers.

[Hahn & Vozila$^+$ 12] S. Hahn, P. Vozila, M. Bisani: Comparison of Grapheme-to-Phoneme Methods on Large Pronunciation Dictionaries and LVCSR Tasks. In *Proc. Int. Conf. on Speech Communication and Technology (Interspeech)*, Portland, OR, USA, Sept. 2012. Electronic, no page numbers.

[Heigold 10] G. Heigold: *A Log-Linear Discriminative Modeling Framework for Speech Recognition*. Ph.D. thesis, RWTH Aachen University, Aachen, Germany, June 2010.

[Heigold & Dreuw$^+$ 10] G. Heigold, P. Dreuw, S. Hahn, R. Schlüter, H. Ney: Margin-based Discriminative Training for String Recognition. *IEEE Journal of Selected Topics in Signal Processing - Statistical Learning Methods for Speech and Language Processing*, Vol. 4, No. 6, pp. 917 – 925, Dec. 2010.

[Heigold & Lehnen$^+$ 08] G. Heigold, P. Lehnen, R. Schlüter, H. Ney: On the Equivalence of Gaussian and Log-Linear HMMs. In *Proc. Int. Conf. on Speech Communication and Technology (Interspeech)*, pp. 273 – 276, Brisbane, Australia, Sept. 2008. ISCA best student paper award.

[Heigold & Ney$^+$ 11] G. Heigold, H. Ney, P. Lehnen, T. Gass, R. Schlüter: Equivalence of Generative and Log-Linear Models. *IEEE Trans. on Audio, Speech and Language Processing*, Vol. 19, No. 5, pp. 1138 – 1148, July 2011.

[Heigold & Schlüter$^+$ 09] G. Heigold, R. Schlüter, H. Ney: Modified MPE/MMI in a Transducer-Based Framework. In *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, pp. 3749 – 3752, Taipei, Taiwan, April 2009.

[Hermansky 90] H. Hermansky: Perceptual Linear Predictive (PLP) Analysis of Speech. *Journal of the Acoustical Society of America*, Vol. 87, No. 4, pp. 1738 – 1752, June 1990.

[Hermansky & Ellis$^+$ 00] H. Hermansky, D. Ellis, S. Sharma: Tandem Connectionist Feature Stream Extraction for Conventional HMM Systems. In *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, pp. 1635 – 1638, Istanbul, Turkey, June 2000.

[Hixon & Schneider$^+$ 11] B. Hixon, E. Schneider, S.L. Epstein: Phonemic Similarity Metrics to Compare Pronunciation Methods. In *Proc. Int. Conf. on Speech Communication and Technology (Interspeech)*, Florence, Italy, Aug. 2011. Electronic, no page numbers.

[Hon & Lee 91] H.W. Hon, K.F. Lee: Recent Progress in Robust Vocabulary-Independent Speech Recognition. In *Proc. DARPA Speech and Natural Language Processing Workshop*, pp. 258 – 263, Pacific Grove, CA, USA, Feb. 1991.

[Huang & Jack 89] X.D. Huang, M.A. Jack: Semi-Continuous Hidden Markov Models for Speech Signals. *Computer Speech and Language*, Vol. 3, No. 3, pp. 239 – 251, 1989.

[Hutchins 13] J. Hutchins: Machine Translation Archive, 2013. http://www.mt-archive.info/.

[Hutchins & Somers 92] W.J. Hutchins, H.L. Somers: *An Introduction to Machine Translation*. Academic Press, Cambridge, 1992.

[IBM 54] IBM: 701 Translator. Press Release, Jan. 1954.

[Jackendoff 90] R. Jackendoff: *Semantic Structures*. MIT Press, Cambridge, MA, 1990.

[Jelinek 69] F. Jelinek: A Fast Sequential Decoding Algorithm Using a Stack. *IBM Journal of Research and Development*, Vol. 13, pp. 675 – 685, Nov. 1969.

[Jelinek 76] F. Jelinek: Continuous Speech Recognition by Statistical Methods. *Proceedings of the IEEE*, Vol. 64, No. 10, pp. 532 – 556, April 1976.

[Jeong & Geunbae Lee 08] M. Jeong, G. Geunbae Lee: Triangular-Chain Conditional Random Fields. *IEEE Trans. on Audio, Speech and Language Processing*, Vol. 16, No. 7, pp. 1287 – 1302, Sept. 2008.

[Ji & Bilmes 06] G. Ji, J. Bilmes: Backoff Model Training using Partially Observed Data: Application to Dialog Act Tagging. In *Proc. Conf. of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pp. 280 – 287, New York City, USA, June 2006. Association for Computational Linguistics.

[Jiampojamarn & Cherry$^+$ 10] S. Jiampojamarn, C. Cherry, G. Kondrak: Integrating Joint n-gram Features into a Discriminative Training Framework. In *Proc. Conf. of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, HLT '10, pp. 697 – 700, Stroudsburg, PA, USA, June 2010. Association for Computational Linguistics.

[Jiampojamarn & Kondrak$^+$ 07] S. Jiampojamarn, G. Kondrak, T. Sherif: Applying Many-to-Many Alignments and Hidden Markov Models to Letter-to-Phoneme Conversion. In *Proc. Conf. of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pp. 372 – 379, Rochester, NY, USA, April 2007.

[Jiampojamarn & Kondrak 09] S. Jiampojamarn, G. Kondrak: Online Discriminative Training for Grapheme-to-Phoneme Conversion. In *Proc. Int. Conf. on Speech Communication and Technology (Interspeech)*, pp. 1303–1306, Brighton, U.K., Sept. 2009.

[Jiang & Hon$^+$ 97] L. Jiang, H.W. Hon, X. Huang: Improvements on a Trainable Letter-to-Sound Converter. In *Proc. European Conf. on Speech Communication and Technology*, Vol. 2, pp. 605 – 608, Rhodes, Greece, Sept. 1997.

[Kanthak & Ney 03] S. Kanthak, H. Ney: Multilingual Acoustic Modeling Using Graphemes. In *Proc. European Conf. on Speech Communication and Technology*, Vol. 1, pp. 1145 – 1148, Geneva, Switzerland, Sept. 2003.

[Kanthak & Schütz$^+$ 00] S. Kanthak, K. Schütz, H. Ney: Using SIMD Instructions for Fast Likelihood Calculation in LVCSR. In *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, pp. 1531 – 1534, Istanbul, Turkey, June 2000.

[Katz 87] S.M. Katz: Estimation of Probabilities from Sparse Data for the Language Model Component of a Speech Recognizer. *IEEE Trans. on Speech and Audio Processing*, Vol. 35, pp. 400 – 401, March 1987.

[Kienappel & Kneser 01] A. Kienappel, R. Kneser: Designing Very Compact Decision Trees for Grapheme-to-Phoneme Transcription. In *Proc. Int. Conf. on Speech Communication and Technology (Interspeech)*, pp. 1911 – 1914, Aalborg, Denmark, Sept. 2001.

[Kneser 00] R. Kneser: Grapheme-to-Phoneme Study. Technical Report WYT-P4091/00002, Philips Speech Processing, Germany, pp. 1 – 23, Nov. 2000.

[Knight 99] K. Knight: Decoding Complexity in Word-Replacement Translation Models. *ACL Computational Linguistics*, Vol. 25, No. 4, pp. 607 – 615, Dec. 1999.

[Koehn 10] P. Koehn: *Statistical Machine Translation*. Cambridge University Press, New York, NY, USA, 2010.

[Koehn 13] P. Koehn: Statistical Machine Translation, 2013. http://www.statmt.org/.

[Koehn & Och$^+$ 03] P. Koehn, F.J. Och, D. Marcu: Statistical Phrase-Based Translation. In *Proc. Conf. of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pp. 127 – 133, Edmonton, Canada, May 2003.

[Koo & Collins 05] T. Koo, M. Collins: Hidden-Variable Models for Discriminative Reranking. In *Proc. Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pp. 507 – 514, Morristown, NJ, USA, Oct. 2005. Association for Computational Linguistics.

[Kozielski & Rybach$^+$ 13] M. Kozielski, D. Rybach, S. Hahn, R. Schlüter, H. Ney: Open Vocabulary Handwriting Recognition Using Combined Word-level and Character-level Language Models. In *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, Vancouver, Canada, May 2013. Electronic, no page numbers.

[Kudo 05] T. Kudo: CRF++ toolkit. http://crfpp.sourceforge.net/, 2005.

[Kudo & Matsumoto 01] T. Kudo, Y. Matsumoto: Chunking with Support Vector Machines. In *Proc. Conf. of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pp. 1 – 8, Morristown, NJ, USA, June 2001. Association for Computational Linguistics.

[Kudo & Matsumoto 05] T. Kudo, Y. Matsumoto: Yet Another Multipurpose CHunk Annotator - YamCha. http://chasen.org/ taku/software/yamcha/, 2005.

[Kumar & Byrne 05] S. Kumar, W. Byrne: Local Phrase Reordering Models for Statistical Machine Translation. In *Proc. Conf. of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pp. 161 – 168, Vancouver, Canada, Oct. 2005.

[Lafferty & McCallum$^+$ 01] J. Lafferty, A. McCallum, F. Pereira: Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proc. Int. Conf. on Machine Learning*, pp. 282 – 289, Williamstown, MA, USA, June 2001.

[Lagarda & Alabau$^+$ 09] A.L. Lagarda, V. Alabau, F. Casacuberta, R. Silva, E. Díaz-de Liaño: Statistical Post-Editing of a Rule-Based Machine Translation System. In *Proc. Conf. of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, NAACL-Short '09, pp. 217 – 220, Boulder, CO, USA, June 2009. Association for Computational Linguistics.

[Lavergne & Cappé$^+$ 10] T. Lavergne, O. Cappé, F. Yvon: Practical Very Large Scale CRFs. In *Proc. Annual Meeting of the Association for Computational Linguistics*, pp. 504 – 513, Uppsala, Sweden, July 2010. Association for Computational Linguistics.

[Lavergne & Crego$^+$ 11] T. Lavergne, J.M. Crego, A. Allauzen, F. Yvon: From n-gram-based to CRF-based Translation Models. In *Proc. of the EACL Int. Workshop on Statistical Machine Translation*, pp. 542 – 553, Edinburgh, Scotland, July 2011.

[Lefèvre 06] F. Lefèvre: A DBN-based Multi-Level Stochastic Spoken Language Understanding System. In *Proc. IEEE Spoken Language Technology Workshop*, pp. 82 – 85, Aruba, Dec. 2006.

[Lefèvre 07] F. Lefèvre: Dynamic Bayesian Networks and Discriminative classifiers for multi-stage semantic interpretation. In *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, Vol. 4, pp. 13 – 16, Honolulu, HI, USA, April 2007.

[Leggetter & Woodland 95] C.J. Leggetter, P.C. Woodland: Maximum Likelihood Linear Regression for Speaker Adaptation of Continuous Density Hidden Markov Models. *Computer Speech and Language*, Vol. 9, No. 2, pp. 171 – 185, 1995.

[Lehnen & Hahn$^+$ 09] P. Lehnen, S. Hahn, H. Ney, A. Mykowiecka: Large-Scale Polish SLU. In *Proc. Int. Conf. on Speech Communication and Technology (Interspeech)*, pp. 2723 – 2726, Brighton, UK, Sept. 2009.

[Lehnen & Hahn$^+$ 11a] P. Lehnen, S. Hahn, A. Guta, H. Ney: Incorporating Alignments into Conditional Random Fields for Grapheme to Phoneme Conversion. In *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, pp. 4916 – 4919, Prague, Czech Republic, May 2011. Electronic, no page numbers.

[Lehnen & Hahn$^+$ 11b] P. Lehnen, S. Hahn, H. Ney: N-grams for Conditional Random Fields or a Failure-transition Posterior for Acyclic FSTs. In *Proc. Int. Conf. on Speech Communication and Technology (Interspeech)*, Florence, Italy, Aug. 2011.

[Lehnen & Hahn$^+$ 12] P. Lehnen, S. Hahn, V.A. Guta, H. Ney: Hidden Conditional Random Fields with M-to-N Alignments for Grapheme-to-Phoneme Conversion. In *Proc. Int. Conf. on Speech Communication and Technology (Interspeech)*, Portland, OR, USA, Sept. 2012. Electronic, no page numbers.

[Levinson & Rabiner$^+$ 83] S.E. Levinson, L.R. Rabiner, M.M. Sondhi: An Introduction to the Application of the Theory of Probabilistic Functions of a Markov Process to Automatic Speech Recognition. *Bell System Technical Journal*, Vol. 62, No. 4, pp. 1035 –1074, April 1983.

[Lowerre 76] B. Lowerre: *A Comparative Performance Analysis of Speech Understanding Systems*. Ph.D. thesis, Carnegie Mellon University, Pittsburgh, PA, USA, 1976.

[Macherey 09] K. Macherey: *Statistical Methods in Natural Language Understanding and Spoken Dialogue Systems*. Ph.D. thesis, RWTH Aachen University, Aachen, Germany, Sept. 2009.

[Macherey & Haferkamp$^+$ 05] W. Macherey, L. Haferkamp, R. Schlüter, H. Ney: Investigations on Error Minimizing Training Criteria for Discriminative Training in Automatic Speech Recognition. In *Proc. Int. Conf. on Speech Communication and Technology (Interspeech)*, pp. 2133 – 2136, Sept. 2005.

[Manning 11] C.D. Manning: Part-of-Speech Tagging from 97% to 100%: Is It Time for Some Linguistics? In A. Gelbukh, editor, *Computational Linguistics and Intelligent Text Processing*, Vol. 6608 of *Lecture Notes in Computer Science*, pp. 171 – 189. Springer Berlin Heidelberg, 2011.

[Marasek & Gubrynowicz 08] K. Marasek, R. Gubrynowicz: Design and Data Collection for Spoken Polish Dialogs Database. In *Proc. Int. Conf. on Language Resources and Evaluation*, Marrakech, Morocco, May 2008. Electronic, no page numbers.

[Matusov & Zens$^+$ 04] E. Matusov, R. Zens, H. Ney: Symmetric Word Alignments for Statistical Machine Translation. In *Proc. Int. Conf. on Computational Linguistics*, pp. 219 – 225, Geneva, Switzerland, Aug. 2004.

[Mauser & Zens$^+$ 06] A. Mauser, R. Zens, E. Matusov, S. Hasan, H. Ney: The RWTH Statistical Machine Translation System for the IWSLT 2006 Evaluation. In *Proc. Int. Workshop on Spoken Language Translation*, pp. 103 – 110, Kyoto, Japan, Nov. 2006. Best Paper Award.

[McCallum & Freitag$^+$ 00] A. McCallum, D. Freitag, F. Pereira: Maximum Entropy Markov Models for Information Extraction and Segmentation. In *Proc. Int. Conf. on Machine Learning*, pp. 591 – 598, Stanford, CA, USA, June 2000.

[Meurs & Lefèvre$^+$ 09] M.J. Meurs, F. Lefèvre, R. de Mori: Spoken Language Interpretation: On the Use of Dynamic Bayesian Networks for Semantic Composition. In *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, pp. 4773 – 4776, Taipei, Taiwan, April 2009.

[Mikolov & Deoras$^+$ 11] T. Mikolov, A. Deoras, D. Povey, L. Burget, J.H. Černoczký: Strategies for Training Large Scale Neural Network Language Models. In *Proc. IEEE Automatic Speech Recognition and Understanding Workshop*, Waikoloa, HI, USA, Dec. 2011. Electronic, no page numbers.

[Mikolov & Karafiát$^+$ 10] T. Mikolov, M. Karafiát, L. Burget, J.H. Černoczký, , S. Khudanpur: Recurrent Neural Network Based Language Model. In *Proc. Int. Conf. on Speech Communication and Technology (Interspeech)*, pp. 1045 – 1048, Makuhari, Japan, April 2010.

[Miller & Schwartz$^+$ 94] S. Miller, R. Schwartz, R. Bobrow, R. Ingria: Statistical Language Processing using Hidden Understanding Models. In *Proc. Workshop on Human Language Technology*, pp. 278 – 282, Morristown, NJ, USA, March 1994. Association for Computational Linguistics.

[Mohri & Pereira$^+$ 02] M. Mohri, F. Pereira, M. Riley: Weighted Finite-State Transducers in Speech Recognition. *Computer Speech and Language*, Vol. 16, No. 1, pp. 69 – 88, 2002.

[Molau 03] S. Molau: *Normalization in the Acoustic Feature Space for Improved Speech Recognition*. Ph.D. thesis, RWTH Aachen, Aachen, Germany, Feb. 2003.

[Moschitti 06] A. Moschitti: Efficient Convolution Kernels for Dependency and Constituent Syntactic Trees. In *Proc. European Conf. on Machine Learning*, pp. 318 – 329, Berlin, Germany, Sept. 2006.

[Moschitti & Pighin$^+$ 06] A. Moschitti, D. Pighin, R. Basili: Semantic Role Labeling via Tree Kernel Joint Inference. In *Proc. Conf. on Computational Natural Language Learning*, pp. 61 – 68, New York City, NY, USA, June 2006.

[Murphy 02] K. Murphy: *Dynamic Bayesian Networks: Representation, Inference and Learning*. Ph.D. thesis, University of California at Berkeley, Berkeley, California, 2002.

[Mykowiecka & Marasek$^+$ 09] A. Mykowiecka, K. Marasek, M. Marciniak, J. Rabiega-Wiśniewska, R. Gubrynowicz: Annotated Corpus of Polish Spoken Dialogues. In Z. Vetulani, H. Uszkoreit, editors, *Human Language Technology. Challenges of the Information Society*, Vol. 5603 of *Lecture Notes in Computer Science*, pp. 50 – 62. Springer Berlin Heidelberg, 2009.

[Ney 84] H. Ney: The Use of a One-Stage Dynamic Programming Algorithm for Connected Word Recognition. *IEEE Trans. on Speech and Audio Processing*, Vol. 32, No. 2, pp. 263 – 271, April 1984.

[Ney 90] H. Ney: Acoustic Modeling of Phoneme Units for Continuous Speech Recognition. In L. Torres, E. Masgrau, M.A. Lagunas, editors, *Signal Processing V: Theories and Applications, Fifth European Signal Processing Conference*, pp. 65 – 72. Elsevier Science Publishers B. V., Barcelona, Spain, 1990.

[Ney & Aubert 94] H. Ney, X. Aubert: A Word Graph Algorithm for Large Vocabulary Continuous Speech Recognition. In *Proc. Int. Conf. on Spoken Language Processing*, Vol. 3, pp. 1355 – 1358, Yokohama, Japan, Sept. 1994.

[Ney & Essen$^+$ 94] H. Ney, U. Essen, R. Kneser: On Structuring Probabilistic Dependencies in Language Modeling. *Computer Speech and Language*, Vol. 2, No. 8, pp. 1 – 38, 1994.

[Ney & Häb-Umbach$^+$ 92] H. Ney, R. Häb-Umbach, B.H. Tran, M. Oerder: Improvements in Beam Search for 10000-Word Continuous Speech Recognition. In *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, Vol. 1, pp. 9 – 12, San Francisco, CA, USA, March 1992.

[Ney & Martin$^+$ 97] H. Ney, S.C. Martin, F. Wessel: Statistical Language Modeling using Leaving-One-Out. In S. Young, G. Bloothooft, editors, *Corpus Based Methods in Language and Speech Processing*, pp. 1 – 26. Kluwer Academic Publishers, Dordrecht, The Netherlands, 1997.

[Ney & Mergel$^+$ 87] H. Ney, D. Mergel, A. Noll, A. Paeseler: A Data-Driven Organization of the Dynamic Programming Beam Search for Continuous Speech Recognition. In *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, pp. 833 – 836, Dallas, TX, USA, April 1987.

[Ney & Oerder 93] H. Ney, M. Oerder: Word Graphs: An Efficient Interface Between Continuous Speech Recognition and Language Understanding. In *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, Vol. 2, pp. 119 – 122, Minneapolis, MN, USA, April 1993.

[NIST 95] NIST: Speech Recognition Scoring Toolkit (SCTK), 1995. http://www.nist.gov/speech/tools/.

[Novak 11] J. Novak: Phonetisaurus: A WFST-driven Phoneticizer. http://code.google.com/p/phonetisaurus/, 2011.

[Novak & Dixon$^+$ 12] J.R. Novak, P.R. Dixon, N. Minematsu, K. Hirose, C. Hori, H. Kashioka: Improving WFST-based G2P Conversion with Alignment Constraints and RNNLM N-best Rescoring. In *Proc. Int. Conf. on Speech Communication and Technology (Interspeech)*, Portland, OR, USA, Sept. 2012. Electronic, no page numbers.

[Och 03] F.J. Och: Minimum Error Rate Training for Statistical Machine Translation. In *Proc. Annual Meeting of the Association for Computational Linguistics*, pp. 160 – 167, Sapporo, Japan, July 2003.

[Och & Ney 00a] F.J. Och, H. Ney: A Comparison of Alignment Models for Statistical Machine Translation. In *Proc. Int. Conf. on Computational Linguistics*, pp. 1086 – 1090, Saarbrücken, Germany, Aug. 2000.

[Och & Ney 00b] F.J. Och, H. Ney: Improved Statistical Alignment Models. In *Proc. Annual Meeting of the Association for Computational Linguistics*, pp. 440 – 447, Hongkong, China, Oct. 2000.

[Och & Ney 02] F.J. Och, H. Ney: Discriminative Training and Maximum Entropy Models for Statistical Machine Translation. In *Proc. Annual Meeting of the Association for Computational Linguistics*, pp. 295 – 302, Philadelphia, PA, USA, July 2002.

[Och & Tillmann$^+$ 99] F. Och, C. Tillmann, H. Ney: Improved Alignment Models for Statistical Machine Translation. In *Proc. of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pp. 20 – 28, Maryland, MD, USA, June 1999.

[Odell & Valtchev$^+$ 94] J.J. Odell, V. Valtchev, P.C. Woodland, S.J. Young: A One-Pass Decoder Design for Large Vocabulary Recognition. In *ARPA Spoken Language Technology Workshop*, pp. 405 – 410, Plainsboro, NJ, USA, March 1994.

[Ortmanns & Ney 95] S. Ortmanns, H. Ney: An Experimental Study of the Search Space for 20000-Word Speech Recognition. In *Proc. European Conf. on Speech Communication and Technology*, Vol. 2, pp. 901 – 904, Madrid, Spain, Sept. 1995.

[Ortmanns & Ney$^+$ 96] S. Ortmanns, H. Ney, A. Eiden: Language-Model Look-Ahead for Large Vocabulary Speech Recognition. In *Proc. Int. Conf. on Spoken Language Processing*, Vol. 4, pp. 2095 – 2098, Philadelphia, PA, USA, Oct. 1996.

[Ortmanns & Ney$^+$ 97] S. Ortmanns, H. Ney, X. Aubert: A Word Graph Algorithm for Large Vocabulary Continuous Speech Recognition. *Computer Speech and Language*, Vol. 11, No. 1, pp. 43 – 72, Jan. 1997.

[Pagel & Lenzo$^+$ 98] V. Pagel, K. Lenzo, A.W. Black: Letter-to-Sound Rules for Accented Lexicon Compression. In *Proc. Int. Conf. on Spoken Language Processing*, Vol. V, pp. 2015 – 2018, Sydney, Australia, Sept. 1998.

[Papineni & Roukos$^+$ 98] K.A. Papineni, S. Roukos, R.T. Ward: Maximum Likelihood and Discriminative Training of Direct Translation Models. In *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, Vol. 1, pp. 189 – 193, Seattle, WA, USA, May 1998.

[Paul 91] D.B. Paul: Algorithms for an Optimal A^* Search and Linearizing the Search in the Stack Decoder. In *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, Vol. 1, pp. 693 – 696, Toronto, Canada, May 1991.

[Perrin & Ralaivola$^+$ 03] B.E. Perrin, L. Ralaivola, A. Mazurie, S. Bottani, J. Mallet, F. d'Alché Buc: Gene Networks Inference using Dynamic Bayesian Networks. *Bioinformatics*, Vol. 19, No. suppl 2, pp. ii138–ii148, 2003.

[Peshkin & Pfefer$^+$ 03] L. Peshkin, A. Pfefer, V. Savova: Bayesian Nets in Syntactic Categorization of Novel Words. In *Proc. Conf. of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pp. 79 – 81, Edmonton, Canada, May 2003.

[Pieraccini & Levin$^+$ 91] R. Pieraccini, E. Levin, C.H. Lee: Stochastic Representation of Conceptual Structure in the ATIS Task. In *Proc. DARPA Speech and Natural Language Processing Workshop*, pp. 121 – 124, Pacific Grove, CA, USA, Feb. 1991.

[Pitz 05] M. Pitz: *Investigations on Linear Transformations for Speaker Adaptation and Normalization*. Ph.D. thesis, RWTH Aachen University, March 2005.

[Plahl & Kozielski$^+$ 13] C. Plahl, M. Kozielski, R. Schlüter, H. Ney: Feature Combination and Stacking of Recurrent and Non-recurrent Neural Networks for LVCSR. In *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, Vancouver, Canada, May 2013. Electronic, no page numbers.

[Platt 99] J.C. Platt: *Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods*. MIT Press, 1999.

[Powell 77] M. Powell: A Fast Algorithm for Nonlinearly Constrained Optimization Calculations. In G. Watson, editor, *Proceedings of the Biennial Conference on Numerical Analysis*, pp. 144 – 157, Dundee, UK, June 1977. Lecture Notes in Mathematics, Vol. 630. Berlin, Heidelberg, New York: Springer 1978.

[Quattoni & Wang$^+$ 07] A. Quattoni, S. Wang, L.P. Morency, M. Collins, T. Darrell: Hidden Conditional Random Fields. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 29, pp. 1848 – 1852, 2007.

[Rabiner & Juang 86] L. Rabiner, B.H. Juang: An Introduction to Hidden Markov Models. *IEEE ASSP Magazine*, Vol. 3, No. 1, pp. 4 – 16, 1986.

[Rabiner & Schafer 79] L.R. Rabiner, R.W. Schafer: *Digital Processing of Speech Signals*. Prentice-Hall Signal Processing Series, Englewood Cliffs, NJ, USA, 1979.

[Ramasubramansian & Paliwal 92] V. Ramasubramansian, K.K. Paliwal: Fast k-dimensional Tree Algorithms for Nearest Neighbor Search with Application to Vector Quantization Encoding. *IEEE Trans. on Speech and Audio Processing*, Vol. 40, No. 3, pp. 518 – 528, March 1992.

[Ramshaw & Marcus 95] L. Ramshaw, M. Marcus: Text Chunking using Transformation-Based Learning. In *Proceedings of the 3rd Workshop on Very Large Corpora*, pp. 84 – 94, Cambridge, MA, USA, June 1995.

[Raymond & Béchet$^+$ 06] C. Raymond, F. Béchet, R. De Mori, G. Damnati: On the Use of Finite State Transducers for Semantic Interpretation. *Speech Communication*, Vol. 48, No. 3-4, pp. 288 – 304, March-April 2006.

[Raymond & Riccardi 07] C. Raymond, G. Riccardi: Generative and Discriminative Algorithms for Spoken Language Understanding. In *Proc. Int. Conf. on Speech Communication and Technology (Interspeech)*, pp. 1605 – 1608, Antwerp, Belgium, Aug. 2007.

[Riedmiller & Braun 93] M. Riedmiller, H. Braun: A Direct Adaptive Method for Faster Backpropagation Learning: The RPROP Algorithm. In *Proc. IEEE Int. Conf. on Neural Networks*, pp. 586 – 591, San Francisco, CA, USA, March 1993.

[Robinson 95] T. Robinson: BEEP - The British English Example Pronunciation Dictionary, 1995. ftp://svr-ftp.eng.cam.ac.uk/comp.speech/dictionaries/.

[Rodriguez & Riccardi$^+$ 07] K.J. Rodriguez, G. Riccardi, M. Poesio, F. Bechet, G. Damnati, K. Marasek, R. Gubrynowicz, A. Mykowiecka, J. Wisniewska, C. Popovici. *Specifications of the Annotation Protocol for the Data*. LUNA consortium, April 2007. http://www.ist-luna.eu/pdf/LUNA-D1.3.pdf.

[Rubinstein & Hastie 97] Y. Rubinstein, T. Hastie: Discriminative vs Informative Learning. In *Proc. Int. Conf. on Knowledge Discovery and Data Mining*, pp. 49 – 53, Newport Beach, CA, USA, Aug. 1997. AAAI Press.

[Sakoe 79] H. Sakoe: Two-Level DP-Matching - A Dynamic Programming-Based Pattern Matching Algorithm for Connected Word Recognition. *IEEE Trans. on Speech and Audio Processing*, Vol. 27, pp. 588 – 595, Dec. 1979.

[Santafé & Lozano$^+$ 07] G. Santafé, J. Lozano, P. Larranaga: Discriminative vs. Generative Learning of Bayesian Network Classifiers. *Lecture Notes in Computer Science*, Vol. 4724, pp. 453 – 464, 2007.

[Schlüter & Bezrukov$^+$ 07] R. Schlüter, I. Bezrukov, H. Wagner, H. Ney: Gammatone Features and Feature Combination for Large Vocabulary Speech Recognition. In *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, Vol. 4, pp. 649 – 652, Honululu, HI, USA, April 2007.

[Schlüter & Müller$^+$ 99] R. Schlüter, B. Müller, F. Wessel, H. Ney: Interdependence of Language Models and Discriminative Training. In *Proc. IEEE Automatic Speech Recognition and Understanding Workshop*, pp. 119 – 122, Keystone, CO, USA, Dec. 1999.

[Schmid 94] H. Schmid: Probabilistic Part-of-Speech Tagging Using Decision Trees. In *Proc. Int. Conf. on New Methods in Language Processing*, pp. 44 – 49, Manchester, UK, Sept. 1994.

[Schwartz & Austin 91] R. Schwartz, S. Austin: A Comparison of Several Approximate Algorithms for Finding Multiple (*N*-Best) Sentence Hypotheses. In *Proc. IEEE Int. Conf. on*

Acoustics, Speech, and Signal Processing, Vol. 1, pp. 701 – 704, Toronto, Canada, May 1991.

[Schwartz & Chow 90] R. Schwartz, Y.L. Chow: The *N*-Best Algorithm: An Efficient and Exact Procedure for Finding the *N* most likely Sentence Hypotheses. In *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, pp. 81 – 84, Albuquerque, NM, USA, April 1990.

[Sejnowski & Rosenberg 86] T.J. Sejnowski, C.R. Rosenberg: NETtalk: a Parallel Network that Learns to Read Aloud. *Neurocomputing (IJON)*, Vol. 1, 1986.

[Sejnowski & Rosenberg 87] T. Sejnowski, C. Rosenberg: Parallel Networks that Learn to Pronounce English Text. *Complex Systems*, Vol. 1, pp. 145 – 168, 1987.

[Sejnowski & Rosenberg 93] T.J. Sejnowski, C.R. Rosenberg: The NETtalk corpus. ftp://svr-ftp.eng.cam.ac.uk/pub/comp.speech/dictionaries/, 1993.

[Seneff 89] S. Seneff: TINA: A Probabilistic Syntactic Parser for Speech Understanding Systems. In *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, pp. 711 – 714, Glasgow, UK, May 1989.

[Servan & Raymond$^+$ 06] C. Servan, C. Raymond, F. Béchet, P. Nocéra: Conceptual Decoding from Word Lattices: Application to the Spoken Dialogue Corpus MEDIA. In *Proc. Int. Conf. on Spoken Language Processing*, pp. 1614 – 1617, Pittsburgh, USA, Sept. 2006.

[Shannon 48] C.E. Shannon: A Mathematical Theory of Communication. *The Bell System Technical Journal*, Vol. 27, pp. 379 – 423, 623 – 656, July, October 1948.

[Shen & Sarkar$^+$ 03] L. Shen, A. Sarkar, A.K. Joshi: Using LTAG Based Features in Parse Reranking. In *Proc. Conf. of Empirical Methods for Natural Language Processing*, pp. 89 – 96, Sapporo, Japan, July 2003.

[Shen & Sarkar$^+$ 04] L. Shen, A. Sarkar, F.J. Och: Discriminative Reranking for Machine Translation. In *Proc. Conf. of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pp. 177 – 184, Boston, MA, USA, May 2004.

[Sixtus 03] A. Sixtus: *Across-Word Phoneme Models for Large Vocabulary Continuous Speech Recognition*. Ph.D. thesis, RWTH Aachen, Jan. 2003.

[Sokolovska & Lavergne$^+$ 10] N. Sokolovska, T. Lavergne, O. Cappé, F. Yvon: Efficient Learning of Sparse Conditional Random Fields for Supervised Sequence Labeling. *IEEE Journal of Selected Topics in Signal Processing*, Vol. 4, No. 6, pp. 953 – 964, Dec. 2010.

[Steinbiss & Ney$^+$ 93] V. Steinbiss, H. Ney, R. Häb-Umbach, B. Tran, U. Essen, R. Kneser, M. Oerder, H. Meier, X. Aubert, C. Dugast, D. Geller: The Philips Research System for Large-Vocabulary Continuous-Speech Recognition. In *Proc. European Conf. on Speech Communication and Technology*, pp. 2125 – 2128, Berlin, Germany, Sept. 1993.

[Stolcke 02] A. Stolcke: SRILM - An Extensible Language Modeling Toolkit. In *Proc. Int. Conf. on Spoken Language Processing*, pp. 901 – 904, Denver, CO, USA, Sept. 2002.

[Stolcke & Bratt$^+$ 00] A. Stolcke, H. Bratt, J. Butzberger, H. Franco, V.R.R. Gadde, M. Plauche, C. Richey, E. Shriberg, K. Sönmez, F. Weng, J. Zheng: The SRI March 2000 Hub-5 Conversational Speech Transcription System. In *NIST Speech Transcription Workshop*, College Park, MD, USA, May 2000.

[Sundermeyer & Nußbaum-Thom$^+$ 11] M. Sundermeyer, M. Nußbaum-Thom, S. Wiesler, C. Plahl, A. El-Desoky Mousa, S. Hahn, D. Nolden, R. Schlüter, H. Ney: The RWTH 2010 Quaero ASR Evaluation System for English, French, and German. In *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, pp. 2212 – 2215, Prague, Czech Republic, May 2011.

[Sutton & McCallum$^+$ 07] C. Sutton, A. McCallum, K. Rohanimanesh: Dynamic Conditional Random Fields: Factorized Probabilistic Models for Labeling and Segmenting Sequence Data. *Journal of Machince Learning Research*, Vol. 8, pp. 693 – 723, May 2007.

[Sutton & McCallum 10] C. Sutton, A. McCallum: An Introduction to Conditional Random Fields. *ArXiv e-prints*, Vol. 1, Nov. 2010. http://arxiv.org/abs/1011.4088.

[Tibshirani 94] R. Tibshirani: Regression Shrinkage and Selection Via the Lasso. *Journal of the Royal Statistical Society, Series B*, Vol. 58, pp. 267 – 288, 1994.

[Tillmann & Ney 03] C. Tillmann, H. Ney: Word Reordering and a Dynamic Programming Beam Search Algorithm for Statistical Machine Translation. *ACL Computational Linguistics*, Vol. 29, No. 1, pp. 97 – 133, March 2003.

[Tomás & Casacuberta 04] J. Tomás, F. Casacuberta: Statistical Machine Translation Decoding Using Target Word Reordering. In *Structural, Syntactic, and Statistical Pattern Recognition*, Vol. 3138 of *Lecture Notes in Computer Science*, pp. 734 – 743. Springer-Verlag, Lisbon, Portugal, Aug. 2004.

[Tur & De Mori 11] G. Tur, R. De Mori: *Spoken Language Understanding: Systems for Extracting Semantic Information from Speech*. John Wiley & Sons, 2011.

[Tur & Hakkani-Tur$^+$ 10] G. Tur, D. Hakkani-Tur, L. Heck: What is left to be understood in ATIS? In *Proc. IEEE Spoken Language Technology Workshop*, pp. 19 – 24, Berkeley, California, USA, Dec. 2010.

[Tur & Wang$^+$ 13] G. Tur, Y.Y. Wang, D. Hakkani-Tur: TechWare: Spoken Language Understanding (SLU) Resources. *IEEE Signal Processing Magazine*, Vol. 30, No. 3, pp. 187 – 189, May 2013.

[Valente & Vepa$^+$ 07] F. Valente, J. Vepa, C. Plahl, C. Gollan, H. Hermansky, R. Schlüter: Hierarchical Neural Networks Feature Extraction for LVCSR system. In *Proc. Int. Conf. on Speech Communication and Technology (Interspeech)*, pp. 42 – 45, Antwerp, Belgium, Aug. 2007.

[van Deemter & Kibble 00] K. van Deemter, R. Kibble: On Coreferring: Coreference in MUC and Related Anotation Schemes. *ACL Computational Linguistics*, Vol. 26, pp. 629 – 637, 2000.

[van den Bosch & Chen$^+$ 06] A. van den Bosch, S. Chen, W. Daelemans, B. Damper, K. Gustafson, Y. Marchand, F. Yvon: PASCAL Letter-to-Phoneme Conversion Challenge. http://pascallin.ecs.soton.ac.uk/Challenges/PRONALSYL/, 2006.

[Vauquois 68] B. Vauquois: A Survey of Formal Grammars and Algorithms for Recognition and Transformation in Machine Translation. In *International Federation for Information Processing Congress*, Vol. 2, pp. 254 – 260, Edinburgh, UK, Aug. 1968.

[Vintsyuk 71] T.K. Vintsyuk: Elementwise Recognition of Continuous Speech Composed of Words from a Specified Dictionary. *Kibernetika*, Vol. 7, pp. 133 – 143, March 1971.

[Viterbi 67] A. Viterbi: Error Bounds for Convolutional Codes and an Asymptotically Optimal Decoding Algorithm. *IEEE Trans. on Information Theory*, Vol. 13, pp. 260 – 269, 1967.

[Vogel & Ney$^+$ 96] S. Vogel, H. Ney, C. Tillmann: HMM-based Word Alignment in Statistical Translation. In *Proc. Int. Conf. on Computational Linguistics*, Vol. 2, pp. 836 – 841, Aug. 1996.

[Vozila & Adams$^+$ 03] P. Vozila, J. Adams, Y. Lobacheva, R. Thomas: Grapheme to Phoneme Conversion and Dictionary Verification Using Graphonemes. In *Proc. European Conf. on Speech Communication and Technology*, pp. 2469 – 2472, Geneva, Switzerland, Sept. 2003.

[Wang & King 11] D. Wang, S. King: Letter-to-Sound Pronunciation Prediction Using Conditional Random Fields. *IEEE Signal Processing Letters*, Vol. 18, No. 2, pp. 122 – 125, 2011.

[Weaver 55] W. Weaver: Translation. In *Machine Translation of Languages: Fourteen Essays*, pp. 15 – 23, MIT, Cambridge, MA, USA, 1955.

[Wegmann & McAllaster$^+$ 96] S. Wegmann, D. McAllaster, J. Orloff, B. Peskin: Speaker Normalization on Conversational Telephone Speech. In *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, Vol. 1, pp. 339 – 341, Atlanta, GA, USA, May 1996.

[Woodland & Gales$^+$ 97] P. Woodland, M. Gales, D. Pye, S. Young: Broadcast News Transcription Using HTK. In *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, Vol. 2, pp. 719 – 722, Munich, Germany, April 1997.

[Woodland & Povey 02] P.C. Woodland, D. Povey: Large Scale Discriminative Training of Hidden Markov Models for Speech Recognition. *Computer Speech and Language*, Vol. 16, No. 1, pp. 25 – 48, 2002.

[Young 92] S.J. Young: The General Use of Tying in Phoneme Based HMM Recognizers. In *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, Vol. 1, pp. 569 – 572, San Francisco, CA, USA, March 1992.

[Yu & Lam 08] X. Yu, W. Lam: Hidden Dynamic Probabilistic Models for Labeling Sequence Data. In *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence*, pp. 739–745, Chicago, IL, USA, July 2008.

[Yvon 96] F. Yvon: *Prononcer par analogie: motivations, formalisations et évaluations*. Ph.D. thesis, École Nationale Supérieure des Télécommunications (ENST), Paris, France, 1996.

[Zens & Ney 03] R. Zens, H. Ney: A Comparative Study on Reordering Constraints in Statistical Machine Translation. In *Proc. Annual Meeting of the Association for Computational Linguistics*, pp. 144 – 151, Sapporo, Japan, July 2003.

[Zens & Och$^+$ 02] R. Zens, F.J. Och, H. Ney: Phrase-Based Statistical Machine Translation. In *German Conference on Artificial Intelligence*, pp. 18 – 32, Aachen, Germany, Sept. 2002.

[Zou & Hastie 05] H. Zou, T. Hastie: Regularization and Variable Selection via the Elastic Net. *Journal of the Royal Statistical Society, Series B*, Vol. 67, pp. 301 – 320, 2005.

[Zweig 98] G. Zweig: *Speech Recognition with Dynamic Bayesian Networks*. Ph.D. thesis, University of California at Berkeley, Berkeley, California, 1998.