Maximum Entropy Models for Sequences: Scaling up from Tagging to Translation

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-Physiker
Patrick Lehnen
aus Aachen

Berichter: Prof. Dr.-Ing. Hermann Ney
Prof. Dr. rer. nat. François Yvon

Tag der mündlichen Prüfung: 17.5.2017

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 12. April 2018

Dipl.-Phys. Patrick Lehnen

Abstract

Maximum entropy approaches for sequences tagging and conditional random fields in particular have shown high potential in a variety of tasks. The effectiveness of these approaches is verified within this thesis using semantic tagging within natural language understanding as an example. Within this task, decent feature engineering and a tuning of the regularization parameter is sufficient to let conditional random fields be superior to a broad set of competing approaches including support vector machines, phrase-based translation, maximum entropy Markov models, dynamic Bayesian networks, and generatively trained probabilistic finite state transducers. Applying conditional random fields to other tasks in many cases calls for extensions to the original notation. For a multi-level semantic tagging in natural language understanding, constrained search is needed, whereas for grapheme-to-phoneme conversion, the support for a hidden segmentation and huge feature sets is required, and for statistical machine translation a solution for the large input *and* output vocabulary, even larger feature sets, and the hidden alignments have to be found. This thesis presents solutions to all these constraints. The conditional random fields are modeled with finite state transducers to support constraints on the search space. They are extended with hidden segmentation, elastic-net regularization, sparse-forward-backward, pruning in training, and intermediate classes in the output layer. Finally, we will combine all extensions to support statistical machine translation with conditional random fields. The best implementation for statistical machine translation is then based on a refined maximum expected Bleu objective using a similar feature notation and the same RPROP parameter estimation. It differs in a more efficient use of the phrase-based or hierarchical baseline with the help of n-best lists.

Zusammenfassung

Maximum-Entropy-Ansätze für Sequenzen und Conditional Random Fields im Speziellen haben bereits für eine größere Anzahl an Aufgaben im Bereich des maschinellen Lernens ihre Effektivität bewiesen. Als Teil dieser Doktorarbeit wird dies am Beispiel des semantischen Taggens aus dem Teilbereich des Sprachverstehens gezeigt. Dort wird gezeigt, dass eine ausführliche Merkmalsauswahl und eine Einstellung des Regularisierungsparameters ausreicht, um ein System zu bauen, was einer Reihe maschineller Lernverfahren überlegen ist. Als konkurrierende Ansätze wurden Support-Vektor-Maschinen, phrasen-basierte Übersetzung, Maximum Entropy Markov-Modelle, Dynamic Bayesian Networks und statistische Finite State Transducer ausgewählt. Möchte man Conditional Random Fields auf andere Aufgaben anwenden, stößt dieser Ansatz an seine Grenzen. Für semantisches Tagging im Rahmen des Sprachverstehens mit einem Mehr-Ebenen-Tagging werden Einschränkungen im Ausgabevokabular benötigt, für eine Graphem-zu-Phonem Umwandlung müssen die Conditional Random Fields eine implizite Segmentierung und sehr große Merkmalssätze unterstützen und für statistische maschinelle Übersetzung wird eine Lösung für die großen Ein- *und* Ausgabevokabularien, komplexe Umordnungen der Wörter und noch größerer Merkmalssätze benötigt. Diese Doktorarbeit präsentiert Lösungen zu diesen Anforderungen. Die Conditional Random Fields werden mit Hilfe von statistischen Automaten modelliert, was eine Einschränkung des Ausgabevokabulars einfach macht. Dieser Ansatz wird erweitert mit impliziten Segmentierungen, Elastic-Net Regularisierung, sparsen Forward-Backward Berechungnen, Approximationen (Pruning) im Training und zwischengelagerten Klassen in der Ausgabeschicht. Abschließend wird alles zusammengefügt, um mit Conditional Random Fields statistische maschinelle Übersetzung zu modellieren. Die beste Implementierung zur Verbesserung der statistischen maschinellen Übersetzung wird erreicht mit einer Anpassung der Zielfunktion zur Maximierung des Bleu-Erwartungswerts. Dieser Ansatz verendet eine ähnliche Merkmalsextraktion und dieselbe Parameterberechnung anhand des RPROP Verfahrens. Jedoch wurden das phrasenbasierte oder hierarchische Grundsystem besser genutzt, indem n-best Listen für die Näherung des Parametertrainings verwendet werden.

Acknowledgment

First, I would like to thank Professor Hermann Ney for the chance to work in the inspiring and challenging environment of his institute, practice teaching in the assignments to his lectures, his support for my research, the chance to work in international projects, and the opportunity to work on my thesis. I want to thank my second reporter Francois Yvon as well to review this thesis.

Making significant improvements in the research of spoken language processing requires the cooperation of multiple persons. I would like to thank first Stefan Hahn. We worked hard together on the improvements in natural language understanding and grapheme-to-phoneme conversion. The fruitful discussions, pair programming sessions, joint supervision of student workers, and travels made the time in the institute a nice time. Georg Heigold taught me maximum entropy modeling and finite state transducers. In statistical machine translation I got a lot of help and inspiration by Carmen Heger, Jörn Wübker, Stephan Peitz, Arne Mauser, Christian Buck, and Christoph Schmidt. I want to thank my office mates Christian Plahl, Markus Nußbaum-Thom, Muhammad Ali Tahir, and Matthias Huck. We had no research collaboration but the discussions gave very important input to my research. Special thanks to everyone proof-reading this thesis. Thank you Stefan Hahn, Jörn Wübker, and Volker Steinbiss.

Most of my research was made possible by the LUNA project funded by the European Union (FP6-033549) and the Quaero program funded by OSEO, French State agency for innovation. The LUNA project gave me the possibility to have interesting research and meet great people. Christian Raymond, Marco Dinarelli, Frederic Bechet, Fabrice Lefèvre, Renato De Mori, Agnieszka Mykowiecka, and Giuseppe Riccardi all were important collaborators. In the Quaero Program, I had the chance to get together with researchers from LIMSI. I especially want to point out the very interesting collaboration with Alexandre Allauzen, Thomas Lavergne, and Francois Yvon. During my visit, we had very interesting discussions about maximum entropy approaches. These discussions condensed in a detailed comparison of our two conditional random fields systems. Of course, I have to also thank the funding agencies and all persons behind the scenes who made this position possible.

I had the chance to (co-)supervise Andreas Guta and Christian Buck in their diploma thesis, and Martin Riess as student worker. I would like to thank them for their hard work. And, to all the other students I taught in the tutorials of Pattern Recognition / Statistical Classification and in the Proseminar: Thanks for the opportunity to practice teaching.

Researchers do new things. And sometimes even a PhD candidate in computer science does experiments in social research. I would like to thank Professor Ney for his participation in this experiment. He permitted the part time, accepted that I left sometimes meetings at four o'clock, and accepted my parental leaves. This was made possible by the Familien-Service-Büro of RWTH Aachen University, the nurseries of our kids, our parents, Google Calendar, and of course and most important my wife, Sabrina Lehnen. She supported this stressful life between research and kids with calendar notifications letting us switch from one world to the other.

Contents

1. Introduction

First, a number of statistical learning methods are compared on semantic tagging being a part of natural language understanding. Semantic tagging is the structuring and categorization of a spoken or written utterance. Chunks of words are assigned to one category. The set of categories is designed by humans. In a second step additional information is extracted from the semantically tagged sequence. The content of the words assigned to the categories is normalized. Complexity in this task is mainly in the abstraction from sentences formed of large vocabularies to a small set of ambiguous categories/tags. The decision of the span of a tag and the tagging label can rarely be decided from simple mapping rules. More often features, e.g. the current word, a predecessor word, or the predecessor decisions, help in the decision for a tagging. We will apply two maximum entropy models. The globally normalized conditional random fields and maximum entropy Markov models normalized on position level. These models are contrasted with support vector machines, dynamic Bayesian networks, phrase-based translation, and a combination of different generative models implemented as weighted finite state transducers. Experimental results verify that conditional random fields model this task best. From the natural language understanding experiments, we will conclude that maximum entropy models can have outstanding recognition performance but the design of the model (global vs. positional normalization) and the training of the model (we will discuss an extension of the training criterion) have a strong influence on the performance.

Conditional random fields cannot be applied out of the box to other tasks such as statistical machine translation or grapheme-to-phoneme conversion. The level of abstraction described in natural language understanding is similar to statistical machine translation. In statistical machine translation, the information presented in a source sentence is expected to be represented in the target sequence. The difference is that in tagging the information is represented with a, compared to statistical machine translation, small set of tags, while in statistical machine translation the target language is of similar complexity as the source language. In semantic tagging the words are tagged in the same order as they appear in the original source sequence. In statistical machine translation the source and target sequence have to represent the same information, but the rules of constructing a sentence may be drastically different. Simple one-to-one mappings of words, as in natural language understanding, are seldom sufficient. The conditional random fields approach from the first chapter needs to be extended to model this structural mapping. The solution in statistical machine translation is phrase-based translation, i.e. the translation of a chunk of source words translated to a chunk of target words. A chunk in one language is called in this thesis a source- or target-*n-gram*. The combination of a source-n-gram with a target-n-gram may be called *phrase* or *joint-n-gram*. This features are important in the modelling and we need to find solutions to model them in the context of maximum entropy sequence models.

As the number of both source-n-grams and target-n-grams may be very large we can expect a very large number of phrases/joint-n-grams. The number of features challenge the conditional random fields algorithms and we need to find strategies to limit the training algorithms to a set of useful features. And finally, conditional random fields include a summation with respect to the full

hypotheses space to normalize the probabilities. This is the same space as used in a full search. As the number of tags in semantic tagging is low (1-100) and at maximum each word is assigned to one tag the full hypotheses space of semantic tagging can be enumerated in recent computers. However, this is not possible for statistical machine translation. State-of-the-art implementations of statistical machine translation make massive use of pruning. E.g the models are pruned before application, and at each point in the construction of the hypotheses only a small set of hypotheses is expanded. Although these different pruning steps are applied, a state-of-the-art statistical machine translation system takes a long time to generate the pruned set of hypotheses for a set of reference translations used in model training. E.g. running a recent statistical machine translation system on a set of 100k training samples has taken five hours with 50 CPUs. Moreover, the summation over the hypotheses hyper-space has to be done in each iteration and typically 50 to some hundreds of iterations are needed.

Implementing conditional random fields for statistical machine translation raises three challenges:

1. Hidden alignments are needed.
2. Complex feature sets have to be handled.
3. The computation of the normalization has to be approximated in a useful way.

At the same time there was interest in improving grapheme-to-phoneme conversion. Grapheme-to-phoneme conversion is the task of translating a sequence of letters to a sequence of phonemes. In other words, it is the determination of the pronunciation of a word. Implementations of grapheme-to-phoneme conversion are needed in automatic speech recognition and text to speech to create lexicons. Grapheme-to-phoneme conversion already includes two of the three statistical machine translation challenges. Here, monotonic hidden alignments and complex feature sets are needed to gain state-of-the-art results. Different strategies to implement the hidden alignment are evaluated. Finally, an extension of the conditional random fields based on a simple HMM-like structure but with complex overlapping feature sets using the alignment only to define the center of the feature functions is proposed. The complex feature sets include long source-n-grams, long target-n-grams, and the combination of both called *joint-n-grams*. The detailed HMM-like search space and the large number of features can increase the computation time and memory requirements over the limits of the computational framework. Thus, approximations to the normalization and the feature selection need to be studied.

The reductions in computation time and memory are large but still not sufficient for statistical machine translation. Thus, the concept of intermediate classes is evaluated. This permitted to build a conditional random fields translation system without using external methods. This system does not reach the performance of a state-of-the-art system but is able to improve a state-of-the-art system in n-best re-scoring. The experiments on natural language understanding show that it may be beneficial to adapt the model or the training to the special conditions of a task. In a last set of experiments the maximum expected BLEU training is studied.

In the next sections, we will present a summarization to all three tasks (natural language understanding Sections 1.1, grapheme-to-phoneme conversion 1.2, and statistical machine translation 1.3).

1.1 Natural Language Understanding (NLU)

In a human-computer interface, a language conversion has to be accomplished. While humans prefer to communicate via a more or less unstructured language, computers need to be given precise queries. E.g. in the domain of a dialog manager responding to human telephone calls or a personal assistant application in a smartphone like Apple's Siri, the human may state a question "How will be the weather tomorrow?". The written text of this question is of less use to the computer, but a structured query "@weather-query[tomorrow]" could be answered, as it is part of a predefined

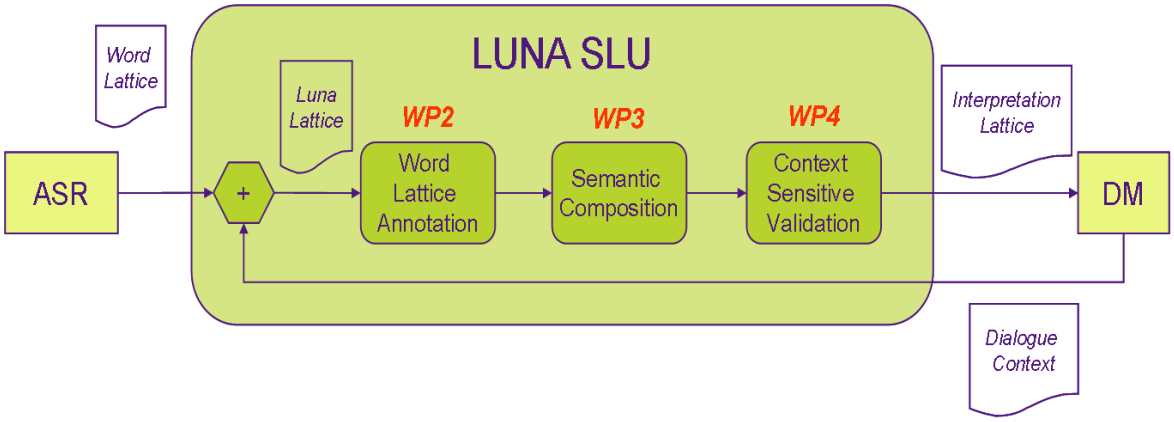


Figure 1.1 The LUNA Pipeline

formal language. Natural language understanding is intended to be the adapter between this two worlds of communication.

From 2006 to 2009, the EU FP6 LUNA project designed a natural language pipeline to process automatic speech recognition (ASR) output and to hand over normalized queries to a dialog manager (DM) (Figure 1.1). The pipeline was designed for three languages, French, Italian, and Polish and included three steps: The *word lattice annotation* or *semantic annotation* (WP2) processes a word sequence $w_1^J = w_1 \dots w_J$ to a sequence of *concepts* $c_1^I = c_1 \dots c_I$, where each of the non overlapping concepts c_i is associated to one or many words $w_{j'}, \dots, w_{j''}$ and describes the information included in these words. This process is more commonly known as *semantic tagging*. The information is described with a context independent formal language defined in the LUNA project. Within this project the formal language was designed for each tackled domain individually (tourist information, public transportation, computer help desk). All languages have in common the decomposition of a concept into an attribute name and an attribute value. Attribute names describe the category of a word sequence, e.g. "weather-query", and attribute values summaries the words in a normalized way, e.g. "tomorrow". In Figure 1.2 an example of the French MEDIA corpus, which is shipped with the LUNA corpora, is presented. "Je veux une chambre double" (original French utterance which can be translated as "I would like a double-bed room") is translated to the concept sequence "@nombre_chambre[1] @chambre_type[double]" (original French annotation which can be translated as "@number_of_rooms[1] @room_type[double]"). The concept sequence may now be further processed with the *semantic composition* module composing *frames* (term from the FrameNet project [Baker & Fillmore$^+$ 98]) as a unit describing the full word sequence in a normalized way including references amongst the concepts, e.g. room(number(1),type(double)). Finally, the *context sensitive validation* disambiguates noise from intended dialog acts. Noise may be unintended calls, unintended start of the application, or fun calls.

Within this thesis, we will focus on semantic tagging ("word lattice annotation" in LUNA). The process of generating concepts sequences as a tuple of attribute name and value sequences from a natural language word sequence.

1.1.1 Error Metric

The quality of semantic tagging is assessed in this thesis with respect to the *concept error rate* (Cer). The Cer is a Levenshtein edit distance based error metric. The Cer is the relation of the minimal number of edit operations *insertion*, *deletion*, and *substitution* needed to convert the hypothesis to the reference and the number of reference concepts.

insertion Add a concept in the hypothesis.

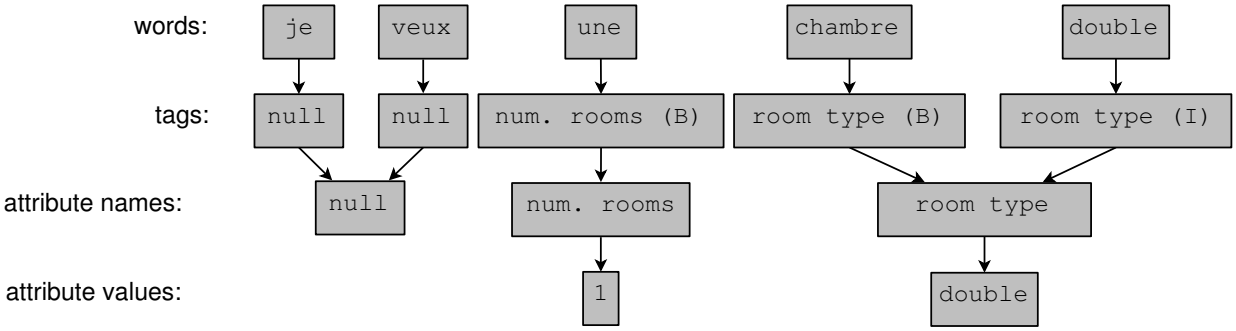


Figure 1.2 Illustrating the use of attribute values with respect to attribute names. (French: "I would like a double-bed room"). The first line shows the input word sequence, the third and fourth line the appropriate attribute names and values (original names in corpus: *nombre-chambre* and *chambre-type*). The second line shows how the 1-to-1 alignment is modeled using "begin" (B) and "inside" (I) tags.

deletion Remove a concept from the hypothesis.
substitution Replace a concept in the hypothesis.
Formally, it is defined as

$$\text{CER} = \frac{\#\text{ insertions} + \#\text{ deletions} + \#\text{ substitutions}}{\#\text{ of reference concepts}}.$$

The number of edit operations can be determined with an effective dynamic programming calculation.

Second metric of choice is the *sentence error rate* SER defined as the ratio of miss-classified utterances and the total number of utterances.

1.1.2 Related Work

The process of preprocessing human speech as input to computer queries is still a new sub-domain of machine learning. The first important corpus for this task was the air travel information service (ATIS) corpus [Price 90, Hemphill & Godfrey$^+$ 90, Dahl & Bates$^+$ 94]. Three humans were placed in a wizard-of-Oz setting, one human having the task to solve travel planning tasks like:

> "Plan the travel arrangements for a small family reunion. First pick a city where the get-together will be held. From 3 different cities (of your choice), find travel arrangements that are suitable for the family members who typify the "economy", "high class", and "adventurous" like style" [Hemphill & Godfrey$^+$ 90]

The user was stating spoken questions to a computer system, which were answered on a screen placed in front of the user. She or he did not know that the answer was not prepared by a computer, but a second human in a different office. While the second human prepared the answer, a third human was transcribing the utterance spoken by the first human. The first human was placed in an office environment with a computer, a monitor, headphones, a head mounted and a desk mounted microphone, a map of the United States, paper, and a pen. The user started the recording by pushing a button. These examples of user-machine dialogs were collected and annotated on different levels, including annotated speech, and concept tags. Over the time more copora have been collected. The overview paper [Tür & Wang$^+$ 13] provides a comprehensive collection including LDC and ELRA catalog numbers.

In 2006, the MEDIA corpus [Bonneau-Maynard & Ayache$^+$ 06] was created as part of a spoken language understanding evaluation campaign. It is a collection of queries asking for hotel room

reservations and general travel information in the south of France to a telephone system in French. The answer component was designed again as wizard-of-Oz. The LUNA project added two additional corpora in Polish and Italian from different domains and extended the MEDIA corpus.

In semantic tagging three main questions arise:

1. What is a suitable preprocessing of the spoken utterance?
2. Which methods to prepare this preprocessing in a suitable and effective way?
3. How is an answer prepared?

The answer to the third question is not part of spoken language understanding anymore, but named *dialog management*, *text generation*, and *text to speech*. The first question can be answered in different ways. Often, a flat description is chosen where the spoken utterance is segmented and a label from a predefined label set is assigned (e.g. the relaxed simplified setting of the MEDIA corpus [Bonneau-Maynard & Ayache$^+$ 06]). Alternatively, and more suitable for a dialog manager is a more formal representation as SQL queries used in the ATIS corpus. Or, sometimes only a short summary like in opinion analysis [Camelin & Béchet$^+$ 10] is needed. Here, an utterance is annotated with the tags *courtesy*, *efficiency*, and *rapidity* with a positive or negative {+,-} polarity expressing the personal experience of a customer at a telephone service of France Telekom. However, the taggers in [Camelin & Béchet$^+$ 10] are applied as flat labeling of segments of words. Only the postprocessing creates a position independent summarization.

The second question is mainly the scope of chapter 3. We focus on a flat labeling of the speech utterance. In [Raymond & Riccardi 07], a first evaluation of different methods to label the MEDIA corpus in the relaxed simplified variant and the ATIS corpus in a similar labeling is conducted. Especially, conditional random fields (CRFs, [Lafferty & McCallum$^+$ 01]), maximum entropy Markov models (MEMMs, [McCallum & Freitag$^+$ 00]), support vector machines (SVMs, [Vapnik 98]), and weighted finite state transducers (FSTs, [Mohri 09, Allauzen & Riley$^+$ 07, Kanthak & Ney 04]) have been extended within the LUNA project resulting e.g. in the final publication [Hahn & Dinarelli$^+$ 11]. Many results in this publication have been created as part of this thesis and can be found in this chapter.

In Parallel and after the LUNA project, Microsoft research conducted research within *spoken query understanding*, which is partially summarized in the tutorial [Li & Tur 11]. An additional interesting recent paper is their tagging of word lattices as described in [Deoras & Tür$^+$ 13]. With applying the positionally normalized MEMMs to word lattices they improved the accuracy of the tagging measured as F-score from 74.8% to 79.2% on their in-house database. Applying conditional random fields on lattices is only possible by approximating the normalization as constant for all utterances, which does not give an improvement in their setup. However, creating an n-best list from the lattices and normalizing for each n-best entry did improve the F-score from 77.1% to 79.0% on their database. Additionally, often a more appropriate ASR output can be chosen from the n-best list / lattice than the first best resulting in a reduced word error rate. Besides accuracy improvements, a more natural way of system reaction is developed e.g. in the thesis [Baumann 13]. Here, all components of the spoken dialog system, the automatic speech recognition, the spoken language understanding, the dialog manager, the text generation, and the final text to speech, work in an incremental way. Already before the last word is spoken the automatic speech recognizer provides words, which it is allowed to change if needed. Same for the following components. So the successor components may process parts of the utterance already before the current component is finished. This opens the possibility of more effective parallelization and to early respond to the input. E.g. the system may respond already before the input is finished. Or, it may respond early after the input is finished. Which would be human-like behaviour.

In this section only an overview could be provided to research related to this thesis. A more detailed introduction of spoken language understanding is given for example in [Tür & Mori 11]. This book is the product of the work of a number of leading spoken language understanding scientists. It includes a chapter covering the history of spoken language understanding, and two

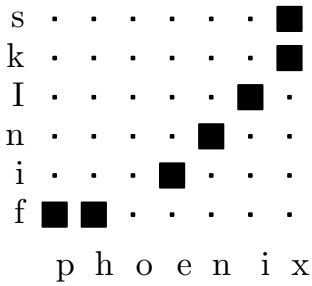


Figure 1.3 Example of an alignment between the grapheme sequence "phoenix" and the phoneme sequence "finIks".

large parts covering human-machine and human-human spoken language understanding. The corpora and tasks used in this thesis are part of human-machine spoken language understanding. Typical tasks in human-human spoken language understanding are summarization, named entity tagging, and speech analysis.

1.2 Grapheme-to-Phoneme Conversion (G2P)

Grapheme-to-phoneme conversion is the task of mapping a lexical word to a sequence of phonemes. This mapping is needed to prepare the word lexicon of an automatic speech recognizer or a text-to-speech system. An automatic speech recognition system decomposes the probability of a word sequence $p(w_1^I|x_1^J)$ into a language model probability $p(w_1^I)$ and an acoustic model probability $p(x_1^J|w_1^I)$, and introduces a log-linearly motivated language model factor α:

$$pr(w_1^I|x_1^J) = \frac{1}{Z} p^{\alpha}(w_1^I) p^{1-\alpha}(x_1^J|w_1^I) \tag{1.1}$$

A calculation of the normalization Z is not needed in search as an argmax operation is applied. The acoustic model does model the probability with respect to a phoneme sequence s_1^K. It is derived with the help of the lexicon of word and phoneme sequence pairs. In general, multiple phoneme sequences for a word sequence are possible. This is solved by the Viterbi approach, taking the phoneme/word sequence pair with maximum probability:

$$x_1^J \mapsto \hat{w}_1^{\hat{I}}(x_1^J) = \underset{w_1^I}{\operatorname{argmax}} \left\{ pr(w_1^I|x_1^J) \right\} = \underset{w_1^I}{\operatorname{argmax}} \left\{ \max_{s_1^K} \left\{ p^{\alpha}(w_1^I) p^{1-\alpha}(x_1^J|s_1^K, w_1^I) \right\} \right\} \tag{1.2}$$

For all sequence pairs not supported by the word lexicon, the acoustic model probability $p(x_1^J|s_1^K, w_1^I)$ is zero. While the language model probabilities are estimated with respect to the word sequence, acoustic model probabilities are estimated with respect to the phoneme sequence.

In principle, word lexicons can be designed by linguists and most commonly a large set of word-to-phoneme mappings is defined by humans. However, there is a large number of words which are missing in the human annotated word lexicons. These could be rare words (e.g. "epithymy"), new words (e.g. "smartphone") or proper names (e.g. "Tahrir square"). Here, grapheme-to-phoneme conversion is needed to extend the word lexicon as needed. From the economic view point, transcribing a lexicon by linguists is a labor-intensive and expensive task. Even as the linguists are asked to design a rule-set to produce the phonemes, this rule-set can be labor-intensive. In [Kominek & Black 06] it has been shown that the number of rules can grow almost linearly in the number of words for some languages, like English, Dutch, or German. So, if only a small amount of correct examples would be sufficient to train a grapheme-to-phoneme converter, the linguist could do more fulfilling tasks.

After Equation (1.2) we stated that the acoustic model probability $p(x_1^J|s_1^K, w_1^I)$ is zero for a word / phoneme sequence pair not in the word lexicon. However, the process of creating words from phoneme sequences can be integrated into the search process. With these new approaches it is possible to reach zero out-of-vocabulary rates with integrating the grapheme-to-phoneme mapping into the automatic speech recognition search space (e.g. [Basha Shaik & Rybach$^+$ 12]).

The words of European languages can mainly be defined by their letter sequence. Only some words need disambiguation from context. Automatic speech recognition can integrate word lexicons with pronunciation variants and the disambiguation can be left to the language model. Thus, it is satisfactory if our grapheme-to-phoneme converter is able to map the correct phoneme sequence as one of the first best translations [Hahn & Lehnen$^+$ 13]. Figure 1.3 describes the grapheme-to-phoneme conversion of the word "phoenix". The black squares define alignment points between the letters on the bottom of the figure and the phonemes on the left side of the figure. Besides the regular mappings of one letter to one phoneme ("e"→"i", "n"→"n", "i"→"I") a mapping of two letters to one phoneme ("ph"→"f"), a mapping of one letter to two phonemes ("x"→"ks"), and a not spoken letter ("o") is included. This example shows the need of many-to-many mappings. Additionally, different to natural language understanding, the corpora rarely include alignment information. Thus, an approach supporting accurate grapheme-to-phoneme conversion has to support hidden many-to-many alignments. For this reason, the conditional random fields implementation in chapter 3 needs to be extended to hidden conditional random fields with monotonous many-to-many alignments in chapter 4.

1.2.1 Error Metric

Grapheme-to-phoneme conversion system outputs are evaluated within this thesis with respect to the *phoneme error rate* (PER) and the *word error rate* (WER). The PER is the relation of Levenshtein edit operations (insertion/deletion/substitution) and the number of reference words. Thus, it is evaluated the same way as the CER in natural language understanding. Furthermore, WER is the relation of hypotheses with at least one error to the number of references. Thus, $1 - \text{WER}$ is the number of perfect conversions. It is not the Levenshtein-based word error rate of an automatic speech recognition.

1.2.2 Related Work

Grapheme-to-phoneme conversion is a core component of automatic speech recognizers and text-to-speech systems. Therefore, a number of baseline techniques are available. An overview can be found in [Hahn & Vozila$^+$ 12] comparing five state-of-the-art converters. In this chapter we have chosen two representative approaches. The approach of [Bisani & Ney 08], representing the best generative baseline, and [Jiampojamarn & Cherry$^+$ 10], a discriminative system leading to the best overall baseline.

The method in [Bisani & Ney 08] is a joint-multigram approach, a development first published in [Deligne & Yvon$^+$ 95]. It segments in parallel the source sequence x_1^J and target sequence y_1^I into K segments of n-grams or *multigrams* $(x_{j_k}^{j_{k+1}-1}, y_{i_k}^{i_{k+1}-1})$ with $i_1 = j_1 = 1$ and $j_{K+1} - 1 = J$, $i_{K+1} - 1 = I$ the length of the source and target sequence. A parallel segment $(x_{j_k}^{j_{k+1}-1}, y_{i_k}^{i_{k+1}-1})$ is called *graphone* in [Bisani & Ney 08]. For instance, the example in Figure 1.3 may be represented as

"phoenix"	=	ph	oe	n	i	x
finIks		f	i	n	I	ks

or

"phoenix"	=	pho	e	ni	x
finIks		f	i	nI	ks

The maximum length of the two n-grams in a graphone is limited by

$$j_{k+1} - j_k \leq L_x$$
$$i_{k+1} - i_k \leq L_y$$

with $L = L_x = L_y$ in [Bisani & Ney 08] and independent numbers in [Deligne & Yvon$^+$ 95]. It has to be noted that this parallel segmentation would evolve together with reordering to the same concept as in phrase based translation [Och & Tillmann$^+$ 99]. In the original publication [Deligne & Yvon$^+$ 95] the joint probabilities $p(x_1^J, y_1^J)$ are estimated as the product of graphone probabilities $p(x_{j_k}^{j_{k+1}-1}, y_{i_k}^{i_{k+1}-1})$ together with bigram probabilities. The graphone probabilities are relative counts estimated from a Viterbi segmentation reestimated in multiple iterations with EM training. Without the bigram probabilities best results were reached with large but similar numbers of L_x and L_y (in the publication $L_x = L_y = 5$). With bigram probabilities the best result was with smaller maximum size of the multigrams ($L_x = 2$, $L_y = 4$). In [Bisani & Ney 08] this approach was extended with an ϵ-symbol on source and target side, but excluding (ϵ, ϵ) graphones. The model was changed to an n-gram model over graphones. N-gram counts were replaced with real-valued *evidence* counts, calculated as the sum of the probability of all paths using this graphone in the search graph spanned by all segmentations respecting the correct source and target sequence divided by the sum of the probability of all paths in the same graph (Baum-Welch). The size of the n-grams is limited by a constant M. To smooth the n-gram probabilities, absolute discounting with discount values λ_n for each level of n-grams is used. These values are tuned by Powell's method on a held out set. Best results were reached if either L or M was high, with the optimal configuration as $L = 1$ and $M \geq 5$. Parameter estimation is with a sum over all segmentations. If search should match this, a sum over all segmentations is needed as well. Doing a maximum over a segmentation sum is not feasible. Thus, in [Bisani & Ney 08] an n-best list was extracted and the probability scores of entries with different segmentation but same hypothesis were summed up and merged to one hypothesis. However, the results seldom changed as there were only few multiple segmentations per hypothesis.

As second baseline [Jiampojamarn & Cherry$^+$ 10] is chosen. It uses the graphone generator presented in [Jiampojamarn & Kondrak$^+$ 07] called *many-to-many alignment* in their publications. It permits the same segmentations as [Bisani & Ney 08] with $L = 2$ and an ϵ-symbol and estimates weights by normalizing the same evidence counts as in [Bisani & Ney 08] called *partial counts* in their publication. The resulting segmentations are used with a phrase based translator to produce n-best lists on the training data, and a support vector machine (SVM) optimizer [Vapnik 98] with a large penalty value for the slack variable is used to simulate a MIRA training [Crammer & Dekel$^+$ 06, Crammer & Singer 03] of a log-linear model. The log-linear model is used to produce a weight $\sum_r \lambda_r h_r(\ldots)$ for each phoneme based on binary features $h_r(\ldots)$ as we use for our maximum entropy approach (Section 4.3). The binary features used are *context* features taking an n-gram from source with maximum length $n \leq 5$ together with the current output phoneme y_i (*source n-grams* in Section 4.3), *transition* features being bigrams on target (*target 2-grams* in Section 4.3), *linear-chain* features as and-combinations of context and transition features, and *joint n-grams* combining context features and transition features with longer history with the same length (e.g. source-3-gram with a target-3-gram, but no source-5-gram with a target-2-gram). Unfortunately, the paper does not explain if the estimates from the many-to-many alignment are used in decoding or training. Theoretically, the estimates could be used in addition to the weights trained with the MIRA training in decoding or they could initialize the λ_r weights before their training. In 2009, the authors of [Jiampojamarn & Cherry$^+$ 10] have been sent the partitioning of the CELEX corpus used in [Bisani & Ney 08]. The results in [Jiampojamarn & Cherry$^+$ 10] are contrasted

with the results in [Bisani & Ney 08]. Thus, we expect the error rates in [Jiampojamarn & Cherry$^+$ 10] to be comparable with [Bisani & Ney 08] and our results on CELEX. Older results, e.g. in [Jiampojamarn & Kondrak$^+$ 07, Jiampojamarn & Kondrak 09] have been generated on a different corpus partition.

1.3 Statistical Machine Translation (SMT)

Machine translation is expected to translate a sequence of words $F = f_1^J = f_1, \ldots, f_J$, $f \in \mathbb{F}$ from one language to a sequence of words in another language $E = e_1^I = e_1, \ldots, e_I$, $e \in \mathbb{E}$, where I is not necessarily equal to J. Both sequences are commonly semantic *sentences*. However, the rules of building a sentence may be different in both languages. So a perfect translation may be from one or many sentences in the source language to one or many sentences in the target language. The vocabularies $\mathbb{F}$, $\mathbb{E}$ are all symbols, which may be seen as *words*. In all languages using Latin letters a word is a sequence of letters associated with some meaning and potentially including grammatical information, e.g. as prefix or suffix. Typical vocabulary sizes $|\mathbb{F}|$, $|\mathbb{E}|$ are in the range of 10,000 to 100,000 words. The words in the source and target sequence do not need to be associated directly, only the *meaning* of both sequences needs to be equal. However, meaning can be interpreted differently by different humans and is hard to measure.

1.3.1 Direct Translation Model

Similar to automatic speech recognition the translation task can be described as a probabilistic problem $p(E|F)$ [Brown & Della Pietra$^+$ 93]. Assuming a 0/1-loss (zero if the sequence E is wrong, and one if sequence E is correct) on the target sequence E the Bayesian decision rule states

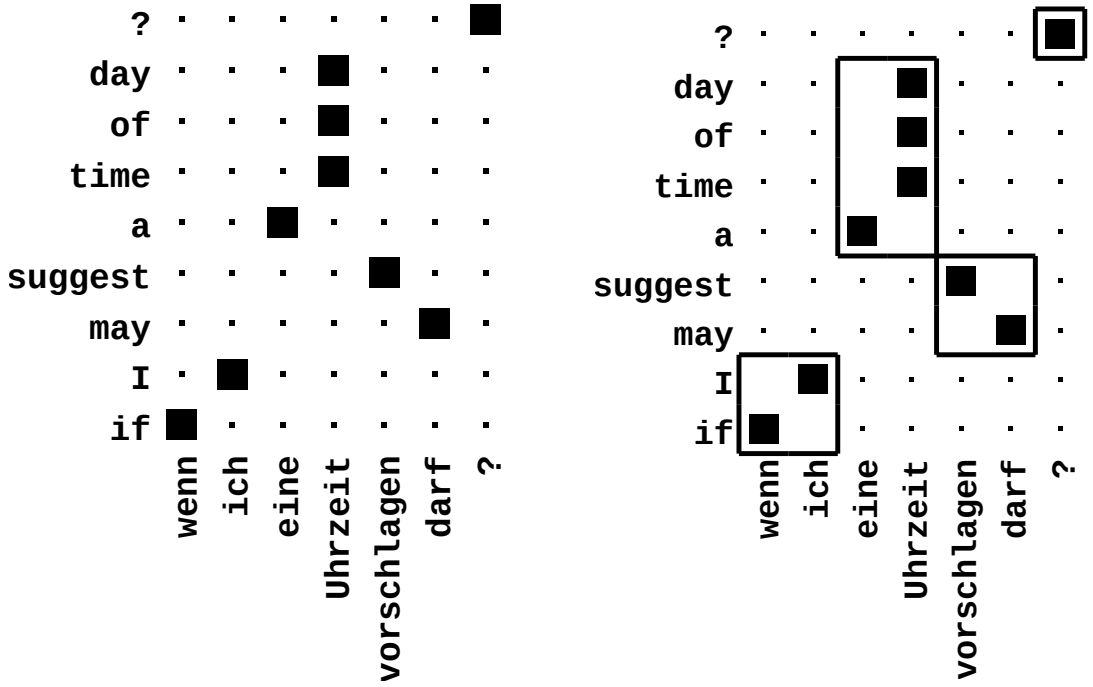
$$\hat{E} = \underset{E}{\operatorname{argmax}} \{p(E|F)\}$$

for the *best* sequence. With the Bayes theorem this statement can be transformed to a factorization of a posterior probability $p(F|E)$ and a prior probability $p(E)$:

$$\hat{E} = \underset{E}{\operatorname{argmax}} \{p(F|E)p(E)\}\,. \tag{1.3}$$

The posterior probability is called translation model and the prior probability is a language model. The language model probability is estimated based on a monolingual set of target language's training sequences. The translation model is estimated based on a parallel bilingual training set composed of pairs of source and target sequences, which are expected to be translations of each other. In [Brown & Della Pietra$^+$ 93] the translation model is a word based model decomposing the posterior probability to a product based on word associations including alignments. A set of five translation models estimated via EM-optimization has been developed (referred to as IBM-1 to -5). These models integrate the estimation of an alignment $a : j \mapsto a_j \in \{0, \ldots, I\}$. The artificial position $a_j = 0$ represents source words without an equivalence in the target sentence. Today, the models IBM-1 to -5 are estimated in source-to-target and target-to-source direction and extended by a hidden Markov model based alignment model [Vogel & Ney$^+$ 96]. The resulting models are used to extract an alignment matrix A for each sentence pair in the bilingual training sets, e.g. with the tool GIZA++ [Och & Ney 03]. An example of the alignment matrix is shown in Figure a. The black boxes represent the estimated alignment points ($A_{ij} \neq 0$) between the German sentence "Wenn ich eine Uhrzeit vorschlagen darf?" and the English sentence "If I may suggest a time of day?".

The IBM-1 to -5 and the hidden Markov model models are a product of probabilities based on one source with one target word and additional alignment and fertility models. However, the



(a) word alignment (b) phrase alignment

Figure 1.4 Example of a word alignment and the corresponding phrase segmentation. (This figure was taken from [Bender 10] by courtesy of the author.)

meaning of one word in one language may translate to a sequence of words in the other language, e.g. "time of day" and "Uhrzeit" in Figure 1.4b. Moreover, sometimes it is only possible to translate a sequence of words to a sequence of words.

1.3.2 Phrase-based Translation

The first statistical translation systems used probabilities modeled with respect to only words f_j, e_i. They were extended in [Och & Tillmann$^+$ 99, Zens & Och$^+$ 02, Koehn & Och$^+$ 03] to phrase-based systems.

Here, the smallest units are called *phrases* defined as tuples of lexical units $\tilde{f}_k = f_{j_k-1}, \ldots, f_{j_k}$. The smallest units modify the search to a construction of a sequence of source-to-target phrase pairs $(\tilde{f}, \tilde{e})_1, \ldots, (\tilde{f}, \tilde{e})_k, \ldots, (\tilde{f}, \tilde{e})_K$ compatible to the given source sequence f_1^J. Words within a phrase need to be contiguous. At the phrase borders a shift within the source sequence is permitted. Generally the size of the shift is limited, as it has a strong effect on the computation time. This implicitly models a *phrase alignment* A of K phrases including the end position of the kth-target phrase i_k, and the begin b_k and end j_k of the kth-source phrase.

$$\ldots, (i_k; b_k, j_k), \ldots$$

Phrase-based translation is similar to a joint-multigram model plus word reordering.

The phrases need to be extracted before search. Here, the alignment points created with the IBM-1 to -5, and hidden Markov models can be used. Certain heuristics extract from the alignment points phrase-based translation units [Och & Tillmann$^+$ 99] called *phrase pairs*. E.g. the *grow-diag-final-and* heuristic [Koehn & Och$^+$ 03]. A phrase pair is a sequence of one or more source words with a sequence of one or more target words. The words within a phrase are expected to be

1. in the same contiguous order as in the source / target sentence,

2. all words in the source phrase have only alignment points within the target phrase, and
3. all words in the target phrase have only alignment points within the source phrase.

From a bilingual training pair all possible phrases are extracted. They do not need to have a linguistic meaning.

Model and parameter estimation for phrase based translation is done in several steps. Normally, bilingual corpora and larger monolingual corpora in the target language are used for the estimation of the models $h_r(f_1^J, A, e_1^I)$ (A defined by the phrase alignment). Independent from the translation models the language model $p(e_1^I) = \prod_{i=1}^{I} p(e_i|e_{i-\delta}^{i-1})$ is estimated on the monolingual corpus [Chen & Goodman 99], which can be done with the SRI language model tool [Stolcke 02]. Phrase translation probabilities and lexical translation probabilities are estimated for each phrase table entry in source-to-target and target-to-source direction. The probabilities are estimated through relative frequencies based on the alignment derived on the bilingual corpus:

$$\text{phrases:} \quad p(\tilde{f}|\tilde{e}) = \frac{N(\tilde{f}, \tilde{e})}{N(\tilde{e})} \tag{1.4}$$

$$\text{lexicals:} \quad p(f|e) = \frac{N(f, e)}{N(e)} \tag{1.5}$$

In a recent development, the source and target phrases do not need to be fully continuous. Instead, they are allowed to include gaps, which are filled by other phrases. The amount of gaps needs to be limited. This approach is called hierarchical phrase-based translation [Chiang 05]. The phrases with gaps and without gaps build a hierarchical structure, which is called *hierarchical parse-tree* or *derivation*. *Glue rules* are added permitting the regular phrase-based translation search space. Thus, the hierarchical search space includes a regular search space as a subset. A phrase pair, optionally including gaps, is called *rule* in hierarchical phrase-based translation to express the relation to *synchronous context free grammars* [Lewis II & Stearns 68]. Through the use of gaps the number of possible rules and derivations increases drastically. Pruning of models and pruning in search is even more critical.

1.3.3 Log-linear Models

All translation models (word-based, phrase-based, syntax-based) produce potential translations, *hypotheses*. These hypotheses need to be weighted to find the best translation. In Equation (1.3) the Bayes decision rule was derived with respect to a 0/1-loss. Taking into account that agreement over the correctness of a translation is low between humans, a 0/1-loss is too strong. In [Och 03], the factorization into a posterior probability and a prior probability is extended with the help of maximum entropy models to

$$p(E|F) = \frac{\max_A \{\exp\left(\sum_r \lambda_r h_r(E, A, F)\right)\}}{\sum_{\tilde{E}} \sum_{\tilde{A}} \exp\left(\sum_r \lambda_r h_r(\tilde{E}, \tilde{A}, F)\right)} \tag{1.6}$$

with the approximation to the first best alignment (see Figure 1.5). The $h_r(E, A, F)$ are called *model scores*, and are the logarithms of the translation model and the language model. With this formulation additional translation models, e.g. word-based together with phrase-based translation models may be included in source-to-target and target-to-source direction. A typical set of phrase-based translation features are phrase-based and word-based translation models in source-to-target and target-to-source direction, a language model, a phrase penalty, and a word penalty. Applying Equation (1.6) in search leaves only the sum of the model scores

$$\hat{E} = \operatorname*{argmax}_E \left\{ \frac{\max_A \{\exp\left(\sum_r \lambda_r h_r(E, A, F)\right)\}}{\sum_{\tilde{E}} \sum_{\tilde{A}} \exp\left(\sum_r \lambda_r h_r(\tilde{E}, \tilde{A}, F)\right)} \right\} = \operatorname*{argmax}_E \left\{ \max_A \left\{ \sum_r \lambda_r h_r(E, A, F) \right\} \right\} \tag{1.7}$$

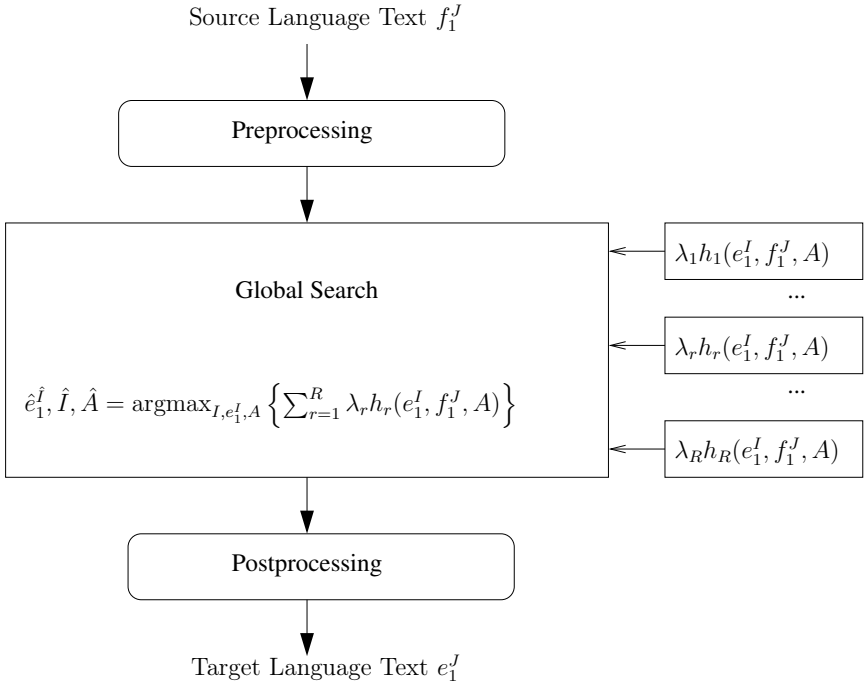


Figure 1.5 Architecture of a log-linear combined translation system.)

as all other parts of the argument of the argmax do not change the argmax. The model parameters λ_1^R are found with the *minimum error rate training* of [Och 03] by extracting an n-best list or a word graph with an initial $\lambda_{1,0}^R$, which is updated iteratively with a line search algorithm for optimal λ_1^R on the n-best list or word graph. *Optimality* is defined by a metric measuring the error rate of the first best translation.

1.3.4 Error Metrics

The most popular metric currently is BLEU [Papineni & Roukos$^+$ 02]. BLEU evaluates accuracy at the document level $(E, R)_1^K$ with E_k the kth-hypothesis and R_k the kth-reference in the evaluation set having $1 \leq k \leq K$ hypotheses and references. It counts n-grams being both in hypothesis E_k and the respective reference R_k and sum them up over k to C(n-grams matched) and relates the result to the number of n-grams in the hypothesis R_k summed up over k to C(n-grams). Each n-gram match between reference and hypothesis as summand in C(n-grams matched) is only counted once, e.g. if the reference includes the unigram "the" only two times in the reference sample R_k it can only be counted two times in the hypothesis E_k. The counts C(n-grams matched) and C(n-grams) are combined to a clipped precision (clipped because of the limitation to count each match only once):

$$p_n = \frac{C(\text{n-grams matched})}{C(\text{n-grams})}. \tag{1.8}$$

These clipped perplexities are log-linearly combined up to 4-grams together with a brevity penalty $BP(l_e, l_r)$ to

$$\text{BLEU}(E, R) = BP(l_e, l_r) \cdot \exp\left(\frac{1}{4}\sum_{n=1}^{4} \log\left(p_n\right)\right) \tag{1.9}$$

with l_e the total length of hypotheses $l_e = \sum_k |E_k|$ and l_r the total length of references $l_r = \sum_k |R_k|$. The brevity penalty is defined as a piecewise function

$$BP(l_e, l_r) = \begin{cases} 1, & \text{if } l_e > l_r \\ \exp\left(1 - l_r/l_e\right), & \text{if } l_e \leq l_r \end{cases} . \tag{1.10}$$

The brevity penalty is intended to compensate that for short hypothesis C(n-grams matched) and C(n-grams) would coverage to each other, letting p_n approach one. As inter-translator agreement is low in translation often multiple references are allowed. With multiple references the matches summed up to C(n-grams matched) are allowed to match between the hypothesis and any reference for the sample k. The total reference length l_r is summed either over the lengths with the shortest distance to the length of the respective hypothesis (best match) or by an average of the reference lengths. In our experiments we used the best matching reference length.

Even if Bleu is a good approximation to human judgment there are examples (e.g. Systran in Figure 4 of [Callison-Burch & Osborne$^+$ 06]) where Bleu fails to correctly rank multiple system outputs.

A different error measure is the *translation edit rate* (Ter, [Snover & Dorr$^+$ 06]). Its design was guided by the idea to count the number of edit operations needed to change the hypotheses into the references. As with Levenshtein-based error measures, it permits the edit operations insertion, deletion, and substitution. And, it adds, compared to Levenshtein-based error measures, a shift edit operation.

insertion Add a word in the hypothesis.
deletion Remove a word from the hypothesis.
substitution Replace a word in the hypothesis.
shift Move a *contiguous* sequence of words within the hypothesis.

The Ter is defined as:

$$\text{Ter} = \frac{\#\text{ insertions} + \#\text{ deletions} + \#\text{ substitutions} + \#\text{ shifts}}{\text{average }\#\text{ of reference words}}$$

Although, the definition of Ter is quite simple, its implementation is not. Other than Levenshtein edit operations the calculation including a shift operation is NP-complete [Shapira & Storer 02]. Thus, not all possible shift operations are tried. The implementations approximate the Ter by applying a heuristic that uses shift operations and calculate the amount of the exact remaining edit operations with dynamic programming. In [Snover & Dorr$^+$ 06], the correlation of one reference Ter to human judgement was estimated to be the same as the correlation of a four reference Bleu (0.39 on MTEval 2004). The correlation of a four reference Ter was significantly larger than the four reference Bleu (0.54 on MTEval 2004).

In this thesis, statistical machine translation experiment results are reported with respect to Bleu and Ter. One of the automatic evaluation metrics or a combination of both is used to automatically optimize the system *and* evaluate the final output.

1.3.5 Related Work

Maximum entropy models have shown high accuracies over a number of tasks including natural language understanding (Chapter 3), grapheme-to-phoneme conversion (Chapter 4), automatic speech recognition [Zweig & Nguyen 09], or language modeling [Roark & Saraclar$^+$ 04]. This success suggests that maximum entropy models should be useful in statistical machine translation, too. A large number of approaches were published, making discriminative training one of the currently most active research areas in statistical machine translation. These approaches propose to extend different places in the system architecture (Figure 1.5). A discriminative model can be

trained independently and added as an additional model to Equation (1.6) (an additional right box in Figure 1.5), or the log-linear combination from Equation (1.6) can be interpreted directly as a maximum entropy model (central box in Figure 1.5). A different classification of approaches is the data taking into account in model parameter estimation. Model parameters changing the log-linear combination are commonly estimated (*optimized*) on a *held-out* set / *development* set in the size of about thousand source/target sequence pairs, while independently estimated models are commonly estimated on 100k to 10M source/target sequence pairs (*training*).

The minimum error rate training (MERT) of [Och 03] can be seen as one of the first maximum entropy estimations of model parameters on a development set. However, the line search algorithm used for parameter estimation does not perfectly fit into the maximum entropy theory as it makes only use of the log-linear combination. The method is not based on derivatives based on the maximum entropy probability estimates. The method of [Och 03] is only able to estimate small sets of parameters (about 10-30). To overcome these restriction the margin infused relaxed algorithm (MIRA, [Crammer & Dekel$^+$ 06]) was applied to model parameter estimation in [Watanabe & Suzuki$^+$ 07, Chiang & Marton$^+$ 08]. MIRA is a large margin training with the constraint that the hypothesis score ranking of an n-best list is equal to the ranking of the n-best with respect to an evaluation metric. In [Watanabe & Suzuki$^+$ 07, Chiang & Marton$^+$ 08] a sentence level approximation to the Bleu [Papineni & Roukos$^+$ 02] is used as ranking metric. Interpreting the ranking of the n-best lists as a binary classification of pairs of hypotheses results in the approach referred to as pairwise ranking optimization (PRO) published in [Hopkins & May 11]. There, a set of pairs from the n-best lists is taken. Each pair is ranked with respect to an evaluation metric and a binary classifier is trained with the higher ranking sentence in class $+1$ and the lower ranking sentence in class -1. In this publication the Bleu+1 sentence level approximation of Bleu is applied. The approach of PRO is further improved in [Green & Wang$^+$ 13] with the use of stochastic gradient descent and AdaGrad instead of large margin training resulting in improvements of up to 2% Bleu absolute. All three approaches MERT, MIRA, and PRO are designed to optimize classification with respect to some evaluation metric. The log-linear nature is only used to combine the models with a weighted sum, but normalization is never calculated. Thus, the final hypothesis weights can not been interpreted as probabilities or logarithms of probabilities.

Alternatively, to the final model parameter estimation (*optimization*), discriminative training may be applied to the creation of models used as a replacement to the standard models. The first approach of this kind was [Liang & Buchard-Côté$^+$ 06] optimizing with perceptron training based on a monotonic phrase-based translation system. It improved with respect to a monotonic phrase-based translation system by 0.8%-Bleu absolute. However, the second reported phrase-based translation system supporting reordering reached about the same result. Two later publications try to adapt conditional random fields for statistical machine translation. Using conditional random fields in their original notation to create these models is infeasible. Their search space is drastically to large as it scales with the size of the largest target-n-gram, even when the idea of sparse-forward-backward (Section 4.6.2) is applied. Thus, approaches try to approximate the model estimations by manually restricting the creation of the search space. E.g. in [Blunsom & Cohn$^+$ 08] a hierarchical phrase-based baseline is used to create the search space constrained by beam width. This baseline is updated with conditional random fields feature estimation with the help of L-BFGS with respect to this constrained search space. In a similar way [Lavergne & Allauzen$^+$ 11] use the n-gram based approach [Casacuberta & Vidal 04, Mariño & Banchs$^+$ 06] to model the reordering, phrase alignment, and the language model. Conditional random fields are applied to estimate the phrase weights. Model updates are carried out by the RPROP algorithm [Riedmiller & Braun 93]. Here, the search space is constrained with a reduced phrase table. However, both approaches only improve over constrained baselines.

Differently, an independently trained discriminative model may be added as an additional model in the log-linear combination of Equation (1.6). In [Mauser & Hasan$^+$ 09], the authors propose

to use an additional lexicon model estimated as the conditional probability of a word being used or not in the target sequence. They report to have 0.8% BLEU and 0.4% BLEU improvement in translation of Chinese and Arabic to English. And, in another line of work, the authors propose to use neural network models to estimate language models [Allauzen & Bonneau-Maynard$^+$ 11, Le & Oparin$^+$ 11] and translation models [Le & Allauzen$^+$ 12] as independent models in Equation (1.6). Here, a concept introduced in [Goodman 01] and originally adapted to neural network language models used in automatic speech recognition by [Morin & Bengio 05] is used. Following [Goodman 01], a conditional probability $p(e|f)$ is decomposed with intermediate *classes* c as $p(e|f) = \sum_c p(e|f,c)p(c|f)$. With language models the variable f is replaced by the history. As reported in [Le & Allauzen$^+$ 12] using neural networks as language model has given 1.0% BLEU and as translation model 1.6% BLEU on a English to French translation task.

In a fourth class of approaches are methods where millions of features as part of the translation models are estimated with respect to a baseline set of generative models. The direct translation model 2 of [Ittycheriah & Roukos 07] has a very different search space construction compared to traditional phrase based models [Och & Tillmann$^+$ 99, Zens & Och$^+$ 02, Koehn & Och$^+$ 03]. It uses minimal translation *blocks*. Translation scores are initialized with normalized phrase counts and updated with a maximum entropy estimation ending in 500k-2.8M features estimated on a training set. Improvements are 0.0%, 1.0%, and 1.7% of BLEU absolute on the used test sets. A large amount of publications report improvements of model estimation based on n-best lists extracted on a training set. E.g. [Simianer & Riezler$^+$ 12] used stochastic gradient descent training of pairwise classification similar to PRO. It reached an improvement of 1.7% BLEU absolute compared to a MIRA trained baseline, but lacks a comparison to a traditional MERT optimized baseline. [He & Deng 12] changed to a maximum expected BLEU objective, which is similar to minimum phone error training in automatic speech recognition [Povey & Woodland 02]. The advantage is a statistically defined objective function like with conditional random fields estimation, which is sensible to the specifics of statistical machine translation where references are not clearly defined. As stated earlier it is hard to define whether or not a translation is correct. E.g. correctness can be evaluated with respect to fluency or semantic completeness. A sentence may still be fluent and the same amount of semantics may be translated even if words are reordered or replaced by synonyms. Weighting each hypothesis in an n-best list with their sentence level BLEU approximation gives a high weight to almost correct hypotheses and a low weight for far from correct hypotheses. Additionally, using maximum expected BLEU is an elegant solution in the common problem of choosing an oracle reference if the reference is not included in the n-best list for conditional log-likelihood objectives, e.g. with conditional random fields. In the original publication, [He & Deng 12] used growth transformation to estimate lexical and phrase weights, and extended this work with stochastic gradient descent optimization in [Gao & He 13]. Here, additionally triplets are estimated and different objective functions are compared, showing significantly better results with maximum expected BLEU objective compared to hinge loss, logistic loss, and logarithmic loss. The hinge loss derives to a perceptron training. The logistic loss results into a setting similar to the hinge loss with an update rule taking the probability of a hypothesis into account. And the logarithmic loss is the loss of conditional random fields with an oracle reference in their setup. In both papers, the improvement compared to a strong baseline was a bit more than 1% BLEU absolute.

1.4 Structure of this Document

The remainder of this work is structured as follows. The next chapter states the scientific goals of this work followed by three chapters structuring the work in the domains of natural language understanding, grapheme-to-phoneme conversion, and statistical machine translation. Within Chapter 3, a set of learning algorithms is evaluated on the domain of natural language

understanding. In this context, conditional random fields are extended to support a multi-layer recognition. Chapter 4 is focused on grapheme-to-phoneme conversion. There hidden conditional random fields are adapted to model the segmentation for grapheme-to-phoneme conversion. To get state of the art performance on grapheme-to-phoneme conversion, large target and source n-grams have to be combined. With standard training algorithms this would be computationally too expensive. Thus, the training and search algorithms and the conditional random field model are adapted to reduce computation time and memory demand. In Chapter 5, the conditional random field software developed over the first two chapters is applied and a second similar approach based on maximum expected BLEU is presented. All three chapters are concluded with the scientific contributions in Chapter 8, followed by a detailed breakdown of the contributions of the different authors in Chapter 9, and possible future research directions in Chapter 7.

The document is finished with list of figures, list of tables, and all bibliographic references are listed in the end of this document.

2. Scientific Goals

Maximum entropy approaches for string-to-string translation have become popular in natural language processing tasks over the last decade. A powerful candidate of this class of models are conditional random fields (CRFs) [Lafferty & McCallum$^+$ 01]. After initial success, e.g. on part of speech tagging [Lafferty & McCallum$^+$ 01], chunking [Sha & Pereira 03], speech recognition [Zweig & Nguyen 09], language modeling [Roark & Saraclar$^+$ 04], and name transliteration [Deselaers & Hasan$^+$ 09], the potential of these approaches for other tasks such as natural language understanding, automatic speech recognition, grapheme-to-phoneme conversion, and statistical machine translation was debated. The research in this thesis is committed to exploiting the potential of the maximum entropy approaches for sequences and pushing the boundaries these approaches do have.

A detailed analysis and comparison of conditional random fields on three natural language understanding corpora First, we assess the potential of maximum entropy sequence approaches on natural language understanding. Within natural language understanding, the task of semantic tagging models the translation from natural language to a semantic ontology. On the source side this task has the same complexity as statistical machine translation while the target language has only a restricted vocabulary of about 100 units and only a monotonous segmentation is needed. This opens the possibility to evaluate a variety of approaches on a real-world human language task.

Application of conditional random fields on a two level label recognition task After semantic tagging the source sequence is only categorized with respect to an ontology. The next step is the extraction of a normalized content from each tagged phrase. For open vocabulary tasks as number normalization this can be best accomplished by a programmatic approach like a set of processing rules. However, for a response (*yes/no*), the answer may be more blurred like "ah, ok", "yes", "mhm", "I dislike that choice". In this case we evaluate the potential of statistical approaches.

Combination of rule based and statistical approaches to the attribute value normalization task We expect that on a different set of semantic tags either a programmatic approach would be better (e.g. numbers) and on other sets a statistical approach should be superior. A system combination approach has to be designed taking the nature of semantic tagging into account.

Support for segmentation with hidden conditional random fields for grapheme-to-phoneme conversion On semantic tagging tasks within natural language understanding, classification and segmentation are usually integrated into one modeling approach and supervised training data includes supervised segmentations. Many tasks do not include alignments or segmentations in the supervised corpus material. For these cases, conditional random fields have to be extended to support segmentations as a hidden variable. Adding arbitrary alignments to conditional random fields would make their training infeasible. We design hidden conditional random fields in a way

imposing only minimal constraints on the segmentation. This approach is developed and evaluated on the task of grapheme-to-phoneme conversion.

Relaxing the memory constraints for conditional random fields To reach state of the art performance, approaches for grapheme-to-phoneme conversion need to leverage knowledge from the translation of one or multiple source units to one or multiple target units. This concept is called joint-n-grams in grapheme-to-phoneme conversion and is very similar to phrase-based translation. Already on grapheme-to-phoneme conversion the number of possible joint-n-grams can exceed the memory of a regular computer. The idea of elastic-net regularization is integrated into the RPROP algorithm and a strategy to keep the memory requirement low is developed.

Relaxing the computation time constraints for conditional random fields During the estimation of conditional random fields, a search is performed on all training instances for multiple iterations. This search has a polynomial complexity with the degree of the largest target n-gram. Sparse-forward-backward and beam pruning in training are assessed on grapheme-to-phoneme conversion and statistical machine translation. To support statistical machine translation with the larger output vocabulary, additionally intermediate classes and training on n-best lists is evaluated.

Application to statistical machine translation Statistical machine translation with its huge vocabularies on source and target side and complex alignments is a very complex task. By nature of this task, already the estimation of phrase count models has large memory requirements and the computation time is in the range of hours. Additionally, the reference translation is not as defined as in other tasks. In many cases the correctness of a translation is debated between human translators. This thesis will present two possible solutions to these challenges. The first solution is the combination of all extensions to conditional random fields described so far. The second approach is applying maximum expected Bleu training, which is based on the same exponential family probability function with similar features and the RPROP algorithm for parameter estimation. In contrast to the approach before, it leverages the phrase-based or the hierarchical baseline more efficiently with the use of n-best lists in training. The final approach consistently improves baseline systems.

3. Applying Maximum Entropy Approaches to Natural Language Understanding

Natural language understanding is a broad task including any postprocessing of automatic speech recognition output with the objective to extract information. The type of information may be different depending on the context. In this chapter, we focus on the production of ordered hierarchical semantic tags $(c_{1,n}^I, c_{1,v}^I)$ with two layers $c_n \in \mathbb{C}_n$ and $c_v \in \mathbb{C}_v$ associated to one or many input words w_1^J, $w \in \mathbb{W}$. The task can be decomposed into different parts. The input words need to be segmented into chunks, each chunk needs to be categorized into one category, and the categorization may be augmented by additional information extracted from the input words. E.g. in the input sentence "I'm driving to Cologne main station" the input words "Cologne main station" may be one chunk associated to the category "station" with the normalized information "Cologne/main station". Resulting in the final tagging "station[Cologne/main station]{Cologne main station}". Categorization and segmentation are joined to the task *attribute name extraction*, represented with the statistical variable c_n. Normalization is addressed as the task *attribute value extraction*, represented with the statistical variable c_v. Mathematically, this decomposition is done with the application of the Bayes theorem

$$p(c_{1,n}^I, c_{1,v}^I | w_1^J) = p(c_{1,n}^I | w_1^J) \cdot p(c_{1,v}^I | c_{1,n}^I, w_1^J)\,. \tag{3.1}$$

And in search, both taggings are applied separately

$$w_1^J \mapsto \hat{c}_{1,n}^I(w_1^J) = \underset{c_{1,n}^I, I}{\operatorname{argmax}} \left\{ p(c_{1,n}^I | w_1^J) \right\} \quad \text{and} \tag{3.2}$$

$$w_1^J, \hat{c}_{1,n}^I \mapsto \hat{c}_{1,v}^I(w_1^J, \hat{c}_{1,n}^I) = \underset{c_{1,v}^I, I}{\operatorname{argmax}} \left\{ p(c_{1,v}^I | \hat{c}_{1,n}^I, w_1^J) \right\}\,. \tag{3.3}$$

In attribute name extraction, six different state-of-the-art learning methods are compared: three generative and three discriminative approaches. We focus on conditional random fields, maximum entropy Markov models, and phrase-based translation, while support vector machines, and dynamic Bayesian networks are provided by other authors. As conditional random fields have been shown to be superior for attribute name extraction, we evaluate the potential of conditional random fields on the tasks of attribute value extraction. Here, the set of possible output labels varies from two in the case of a response ("yes", "no") to many hundred possible labels (e.g. times, street names) and constraints have to be implemented to augment the attribute names only with valid attribute values. With these constraints and the largely varying label set sizes the attribute value extraction is very similar to the second layer in the intermediate classes implementation for statistical machine translation (Section 5.1).

3.1 Corpora

Within the research documented in this chapter, three corpora are utilized. The French MEDIA corpus, the Polish LUNA corpus, and Italian LUNA corpus. Their statistics can be compared in table 3.1. All three corpora are assembled of telephone speech recorded in a central call center with varying callers. For all calls, manually annotation as well as automatic speech recognition is available, opening the possibility to compare the stability of the methods to noisy input. Additionally, through varying languages and domains, the corpora cover a wide range of possible inputs.

3.1.1 The French MEDIA Corpus

This corpus was already available at the beginning of the LUNA project and is the reference corpus of most experiments. It was published as a product of the Media/Evalda campaign [Bonneau-Maynard & Ayache$^+$ 06] and covers a touristic domain, including e.g. hotel room reservations. It is collected with human-machine telephone calls in a wizard-of-Oz manner.

An example is the typical hotel room reservation:

"*je veux une chambre double pour deux personnes*" (I would like a double-bed room for two persons).

Which would be assigned to a sequence out of 99 attribute names:

original corpus		translation	
spoken text	attribute name	spoken text	attribute name
je veux	`null`	*I would like*	null
une	`nombre-chambre`	*a*	number of hotel rooms
chambre double	`chambre-type`	*double-bed room*	type of hotel room
pour deux personnes	`sejour-nbPersonne`	*for two persons*	num. of pers. per stay

These chunks of words together with an attribute name are mapped to normalized attribute values. These attribute values are either numeric units, proper names or semantic classes. The final output of the system for the initial input then could be:

spoken text	attribute name	attribute value
je veux	`null`	
une	`nombre-chambre`	`1`
chambre double	`chambre-type`	`double`
pour deux personnes	`sejour-nbPersonne`	`2`

Experiments in this thesis use only the annotation of the MEDIA corpus which is referred to as *relaxed simplified*. This annotation includes attribute names and values and a specifier called mode, which can take values of $\{+, -\}$ in this setting and is concatenated with the attribute name. Additionally, the corpus includes annotations, called *specifiers*. Describing certain relations between attribute names and values. And, it includes annotations distinguishing major speech acts like assertion, negation, and request [Bonneau-Maynard & Ayache$^+$ 06].

3.1.2 The Polish LUNA Corpus

Here, the human-human calls to the call center of the Warsaw transportation system are recorded [Marasek & Gubrynowicz 08, Mykowiecka & Marasek$^+$ 09]. Calls cover questions about routes, itinerary, stops, and fare reductions. With 195 possible attribute names this corpus has the largest attribute name set. Some of the attribute names are closely related and can only be discriminated

with context information. Polish is an inflectional language with a relatively free word order. Morphologic information is likely to be important for all learning approaches.

An example is a request concerning a specific bus number (English translation in parentheses):

"*chciałam linię sto pięćdziesiąt jeden...*" (I would like line one hundred fifty one...)

This request should be represented as:

original corpus		translation	
spoken text	attribute name	spoken text	attribute name
chciałam	`Action`	*I would like*	action
linię sto pięćdziesiąt jeden	`BUS`	*one hundred fifty one*	bus line

Finally processed to:

spoken text	attribute name	attribute value
chciałam	`Action`	`Request`
linię sto pięćdziesiąt jeden	`BUS`	`151`

3.1.3 The Italian LUNA Corpus

The second corpus collected in the LUNA project is the Italian LUNA corpus [Dinarelli & Quarteroni$^+$ 09]. It was collected at the computer help desk of CSI Piemonte, a public regional institution. Like the MEDIA corpus it was collected with human-machine dialogs created in a wizard-of-Oz setting. With 43 attribute names trained with 3,171 utterances it is the smallest of the three corpora.

A typical example is the description of a problem with the printer:

"*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)

For this example the expected tagging is:

original corpus		translation	
spoken text	attribute name	spoken text	attribute name
Buongiorno io ho	`null`	*Good morning I have*	null
un problema	`HardwareProblem .type`	*a problem*	type of hardware problem
con la stampante	`Peripheral.type`	*with the printer*	type of peripheral
da questa mattina	`Time.relative`	*since this morning*	relative time
non riesco	`HardwareOperation .negate`	*cannot*	negate hardware operation
piu'	`null`	*any more*	null
a stampare	`HardwareOperation .operationType`	*print*	type of hardware operation

The corresponding attribute-value annotation is:

spoken text	attribute name	attribute value
Buongiorno io ho	`null`	
un problema	`HardwareProblem.type`	`general problem`
con la stampante	`Peripheral.type`	`printer`
da questa mattina	`Time.relative`	`morning`
non riesco	`HardwareOperation.negate`	`non`
piu'	`null`	
a stampare	`HardwareOperation.operationType`	`to_print`

Table 3.1 Statistics of the training, development and evaluation SLU corpora as used for all experiments.

		Train		Dev		Eval	
		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
	ASR Wer [%]	–		30.3		31.4	
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
	ASR Wer [%]	–		39.5		38.9	
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
	ASR Wer [%]	–		28.5		27.0	

3.2 Design of an Attribute Name Extraction Module (Concept Tagging) within the LUNA Context

Beginning the task of semantic tagging, the first question to be answered is the choice of the machine learning methods to be chosen. Within this section, we will focus on the selection of attribute names, the *category* of a sequence of words, with respect to the spoken utterance represented as plain text provided by an automatic speech recognizer or human transcribed speech.

Many approaches do not support alignments. They commonly only support one output label per input word. The annotation in semantic tagging includes in the used corpora a segmentation of multiple input words to one label. To be able to let the approaches without segmentation support produce a segmentation we adopt the *BIO scheme*, proposed in [Ramshaw & Marcus 95] (Section 3.2.1). We contrast conditional random fields, maximum entropy Markov models, phrase based translation, a generative weighted finite state transducer approach applying a trigram language model, support vector machines, and dynamic Bayesian networks. The methods' designs and parameterizations are described in Section 3.2.2. A large margin extension of conditional random fields is presented in Section 3.2.3. Furthermore, as the design of a maximum entropy sequence model software is part of this contribution, the implementation of conditional random fields is also described in Section 3.2.4. Finally, all methods are contrasted with respect to the three corpora presented in Section 3.1 showing the better performance of conditional random fields on both transcribed speech and automatic speech recognition input (Section 3.2.5).

3.2.1 1-to-1 Alignment

An attribute name $c_{i,n}$ may be associated to one or more words w_j, while a word w_j is associated to exactly one attribute name $c_{i,n}$, which can be described with a segmentation a_1^J. Segmentations are included in the references of the semantic annotation corpora $(\bar{c}_1^{\bar{I}}, \bar{a}_1^J)_n$, with $n = 1, \ldots, N$ running over the corpus utterances. Thus, a model describing $p(c_1^I, a_1^J, I|w_1^J)$ will be needed. With *begin* (B) and *inside* (I) labels the tuple (c_1^I, a_1^J) can be represented as a sequences t_1^J with the same length as the word sequence w_1^J. E.g. the utterance part "chambre double" in Figure 1.2 is mapped to:

$$\underbrace{\text{“chambre”:chambre-type}_{\mathbf{B}\text{egin}}}_{w_4 : t_4}, \underbrace{\text{“double”:chambre-type}_{\mathbf{I}\text{nside}}}_{w_5 : t_5}$$

Here, the *BIO scheme*, proposed in [Ramshaw & Marcus 95], has been adopted.

The models describing $p(t_1^J|w_1^J)$ do not need to explicitly include the alignment/segmentation information, and the problem can be seen as a 1-to-1 aligned monotonous translation. Before the final output of the systems the tags t_1^J are mapped back to the concept sequence together with the segmentation $(c_{1,n}^I, a_1^J)$, and the individual parts of the output can be evaluated.

3.2.2 Models

Six different models are used in attribute name extraction. The three discriminative models conditional random fields, maximum entropy Markov models, and support vector machines are applied together with the three generative models phrase-based translation, weighted finite state transducers, and dynamic Bayesian networks. In Section 3.2.1, the original probability $p(c_1^I, a_1^J, I|w_1^J)$ is converted to a tag mapping problem $p(t_1^J|w_1^J)$, which will be modeled with the different statistical approaches used throughout this section.

3.2.2.1 Conditional Random Fields and Maximum Entropy Markov Models

Conditional random fields (CRFs) [Sutton & McCallum 10] are maximum entropy models normalized over all target sequences t_1^I, conditioned on the source sequence w_1^I, and parameterized with values $\lambda_1^R = \lambda_1, \ldots, \lambda_r, \ldots, \lambda_R$ based on *feature functions* $H_r(t_1^J, w_1^J)$:

$$p_{\lambda_1^R}(t_1^J|w_1^J) = \frac{\exp\left(\sum_{r=1}^R \lambda_r H_r(t_1^J, w_1^J)\right)}{\sum_{\tilde{t}_1^J} \exp\left(\sum_{r=1}^R \lambda_r H_r(\tilde{t}_1^J, w_1^J)\right)}, \tag{3.4}$$

The parameters λ_1^R are estimated with respect to conditional maximum likelihood (with $n = 1, \ldots, N$ training samples $(\hat{t}_{n,1}^{J_n}, w_{n,1}^{J_n})$):

$$\hat{\lambda}_1^R = \underset{\lambda_1^R}{\operatorname{argmax}} \left\{ \underbrace{\sum_{n=1}^N \log\left(p_{\lambda_1^R}(\hat{t}_{1,n}^{J_n}|w_{1,n}^{J_n})\right)}_{L} \right\} \tag{3.5}$$

$$\Leftrightarrow \frac{\partial L}{\partial \lambda_r} = 0$$

In [Berger & Miller 98], a prior $p(\lambda_1^R)$ in the parameters λ_1^R is combined with the conditional log-likelihood estimation L resulting to an extended conditional log-likelihood L':

$$L' = \log\left(p(\lambda_1^R)\right) + L$$

Now choosing $p(\lambda_1^R)$ as

$$p(\lambda_1^R) = p_\nu(\lambda_1^R) = \frac{1}{Z''} \exp\left(-\frac{1}{2} c \|\lambda_1^R\|_\nu^\nu\right) \tag{3.6}$$

with $\nu \in \mathbb{N}$ (e.g. a Gaussian prior with $\nu = 2$) or combinations $p(\lambda_1^R) = p_1(\lambda_1^R) \cdot p_2(\lambda_1^R)$ (called Elastic-Net, [Zou & Hastie 05]) results in

$$\hat{\Lambda} = \underset{\Lambda}{\operatorname{argmax}} \left\{ L + \log\left(p(\Lambda)\right) \right\}$$

The concept of introducing a prior to the parameters is also called *regularization*. The prior $p(\Lambda) = p_2(\Lambda)$ is called *L2 regularization*, while $p(\Lambda) = p_1(\Lambda) \cdot p_2(\Lambda)$ is called *Elastic-Net*.

If the feature functions are constrained to some context, e.g. next neighbor (*Markov Assumption / n-gram*), $\exp\left(H(t_1^J, w_1^J)\right)$ can be factorized to $\prod_{j=1}^J \exp\left(h(t_{j-\delta}^j, w_1^J)\right)$ with $t_{j-\delta}^j = t_{j-\delta}, \ldots, t_j$ and some $\delta \in \mathbb{N}_0$:

$$p(t_1^J|w_1^J) = \frac{\prod_{j=1}^J \exp\left(\sum_{r=1}^R \lambda_r h_r(t_{j-\delta}^j, w_1^J)\right)}{\sum_{\tilde{t}_1^J} \prod_{j=1}^J \exp\left(\sum_{r=1}^R \lambda_r h_r(\tilde{t}_{j-\delta}^j, w_1^J)\right)}, \tag{3.7}$$

which is the original modeling of conditional random fields called *linear chain conditional random fields* (LCCRF) [Lafferty & McCallum$^+$ 01]. As a consequence of this constraint, the computation time of the model parameter estimation (Equation (3.9)) can be reduced with the dynamic programming factorization:

$$\frac{\partial L'}{\partial \lambda_r} = \sum_{n=1}^N H_r(t_{1,n}^J, w_{1,n}^J) - \sum_{n=1}^N p(t_{1,n}^J|w_{1,n}^J) h_r(t_{1,n}^J, w_{1,n}^J) + \frac{\partial \log\left(p(\Lambda)\right)}{\partial \lambda_r} \tag{3.8}$$

$$= \sum_{n=1}^N N_{r,n} - \sum_{n=1}^N D_{r,n} - \frac{\partial \log\left(p(\Lambda)\right)}{\partial \lambda_r} \stackrel{!}{=} 0 \tag{3.9}$$

$$\textit{(Numerator)} \qquad N_{r,n} = \sum_{j=1}^{J} h_r(\hat{t}^j_{n,j-\delta}, w^J_{n,1})$$

$$\textit{(Denominator)} \qquad D_{r,n} = \sum_{j=1}^{J} \sum_{\tilde{t}^j_{j-\delta}} p(\tilde{t}^j_{j-\delta}|w^J_{n,1}) h_r(\tilde{t}^j_{j-\delta}, w^J_{n,1})$$

$N_{r,n}$ is the empirical expectation value of the function h_r with respect to the reference and $D_{r,n}$ is the expectation value of h_r with respect to the estimated model conditioned on $w^J_{n,1}$. The function p (*posterior*) is defined on the basis of the forward accumulator α and the backward accumulator β as

$$\begin{aligned} p(t^j_{j-\delta}|w^J_1) &= \frac{1}{Z}\alpha_j(t^{j-1}_{j-\delta}|w^J_1) \exp\left(h(t^j_{j-\delta}, w^J_1)\right) \beta_{j+1}(t^j_{j-\delta}|w^J_1)\,, \\ \alpha_j(t^{j-1}_{j-\delta}|w^J_1) &= \sum_{t_{j-2}} \exp\left(h(t^{j-1}_{j-1-\delta}, w^J_1)\right) \alpha_{j-1}(t^{j-2}_{j-1-\delta}|w^J_1)\,, \\ \beta_{j+1}(t^j_{j-\delta}|w^J_1) &= \sum_{t_{j+1}} \exp\left(h(t^{j+1}_{j+1-\delta}, w^J_1)\right) \beta_{j+2}(t^{j+1}_{j+1-\delta}|w^J_1)\,, \end{aligned} \tag{3.10}$$

$$\beta_{N+1} = 1, \quad \alpha_1 = 1, \quad Z = \beta_0(\$)\,.$$

The training criterion cannot be solved analytically and need to be solved with numeric optimization methods. In the presented work we will apply the RPROP algorithm [Riedmiller & Braun 93].

Following the Bayes decision rule during search, the best target sequence with respect to a given source sequence is found by

$$\hat{t}^J_1 = \underset{t^J_1}{\operatorname{argmax}} \left\{p(t^J_1|w^J_1)\right\} \overset{\text{CRF}}{=} \underset{t^J_1}{\operatorname{argmax}} \left\{\sum_{j=1}^{J} h(t^j_{j-\delta}, w^J_1)\right\}, \tag{3.11}$$

as the denominator does not depend on the target sequence and the exponential function $\exp()$ is a monotonically increasing function. In attribute name extraction, only bigrams are applied, which is addressed by setting $\delta = 1$. In Chapter 4, the equations above are used with arbitrary δ.

The computation of $N_{r,n}$ is a simple feature count and can be implemented efficiently. The computation of $D_{r,n}$ is much more expensive. Here, a sum over all possible hypothesis n-grams $\tilde{t}^j_{j-\delta}$ is needed.

The *maximum entropy Markov model (MEMM)* [McCallum & Freitag$^+$ 00] factorizes $p(t^J_1|w^J_1)$ with the Bayes theorem similar to language modeling:

$$p(t^J_1|w^J_1) = \prod_{j=1}^{J} p(t_j|t^{j-1}_1, w^J_1) = \prod_{j=1}^{J} p(t_j|t^{j-1}_{j-\delta}, w^J_1), \tag{3.12}$$

where δ is again an δ-neighbor / δ-gram context. $p(t_j|t^{j-1}_{j-\delta}, w^J_1)$ is modeled with an maximum entropy model normalized per position:

$$\begin{aligned} p(t^J_1|w^J_1) = \prod_{j=1}^{J} p(t_j|t^{j-1}_1, w^J_1) &= \prod_{j=1}^{J} \frac{\exp\left(\sum_{r=1}^{R} \lambda_r h_r(t_j, t^{j-1}_{j-\delta}, w^J_1)\right)}{\sum_{\tilde{t}} \exp\left(\sum_{r=1}^{R} \lambda_r h_r(\tilde{t}, t^{j-1}_{j-\delta}, w^J_1)\right)} \\ &= \frac{\prod_{j=1}^{J} \exp\left(\sum_{r=1}^{R} \lambda_r h_r(t_j, t^{j-1}_{j-\delta}, w^J_1)\right)}{\prod_{j=1}^{J} \sum_{\tilde{t}} \exp\left(\sum_{r=1}^{R} \lambda_r h_r(\tilde{t}, t^{j-1}_{j-\delta}, w^J_1)\right)} \end{aligned} \tag{3.13}$$

Which is similar to the probability of conditional random fields Equation (3.7) but with sum interchanged with product in the denominator. Maximum entropy Markov models are estimated with respect to the conditional log-likelihood criterion L', too. This results in the same derivates for $N_{r,n}$ and the regularization part. The denominator part $D_{n,r}$ is:

$$D_{r,n} = \sum_{j=1}^{J} p(\tilde{t}_j|\hat{t}_{n,j-\delta}^{j-1}, w_{n,1}^{J}) h_r(\tilde{t}_i, \hat{t}_{n,j-\delta}^{j-1}, w_{n,1}^{J}) \tag{3.14}$$

Here, a sum over all possible n-grams and a posterior calculation are not needed.

Different to conditional random fields, the normalization is not constant with respect to the search argument and has to be kept in search:

$$\hat{t}_1^J = \underset{t_1^J}{\operatorname{argmax}} \left\{ p(t_1^J|w_1^J) \right\} \overset{\text{MEMM}}{=} \underset{t_1^J}{\operatorname{argmax}} \left\{ \prod_{j=1}^{J} \frac{\exp\left(h(t_j, t_{j-\delta}^{j-1}, w_1^J)\right)}{\sum_{\tilde{t}} \exp\left(h(\tilde{t}, t_{j-\delta}^{j-1}, w_1^J)\right)} \right\} \tag{3.15}$$

In maximum entropy Markov models, the computation time is dominated by the sum over all labels t in the denominator of Equation (3.13). The number of labels is $|\mathbb{T}|$. Thus, the model parameter estimation of maximum entropy Markov models is by a factor of $|\mathbb{T}|^{\delta}$ faster than the model parameter estimation of conditional random fields.

If the feature functions of conditional random fields do not take context into account ($\delta = 0$), the product over $\exp\left(h(t_j, w_1^J)\right)$ in Equation (3.7) can be decomposed. Then, Equation (3.7) and Equation (3.13) are equivalent. Consequently, conditional random fields and maximum entropy Markov models are equivalent in this special case.

Both models are based on feature function $h_r(t_{j-\delta}^{j}, w_1^J)$, which may be any functions of the given source and target symbols. However, throughout this thesis we will specialize on binary feature functions $h_r(t_{j-\delta}^{j}, w_1^J) \in \{0, 1\}$, introducing the opportunity of using efficient modeling of this functions, e.g. as hash-tables. The precise selection of the feature functions depend on the given task. For natural language understanding we use

- source-to-target features:
  1. lexical features $(w_{j+\alpha}, t_j)$, being 1 if and only if a combination of a source symbol with offset α and a target symbol is found, e.g. if w_{j-1} is "the" and t_j is "$\text{name}_{\text{begin}}$" (in NLU: the attribute name "name"),
  2. (word part) prefix features $(\mathit{substr}(w_j, 0, \beta_1), t_j)$, being 1 if and only if a combination of the first β_1 letters of the current source symbol and a target symbol is found, e.g. if $\mathit{substr}(w_j = \text{"euros"}, 0, \beta_1 = 4) = \text{"euro"}$ and t_j is "$\text{currency}_{\text{begin}}$".
  3. (word part) suffix features $(\mathit{substr}(w_j, \beta_2, |w_j| - \beta_2), t_j)$, operating on the last β_2 letters,
  4. capitalization $(\mathit{cap}(w_j), t_j)$, being 1 only if a source symbol starts with a capital letter, and
- target-bigram features (t_{j-1}, t_j):
  being 1 if and only if a combination of a predecessor and a current target symbol is found, e.g. the predecessor attribute name is "number" and the attribute name is "currency". Target-bigrams in natural language understanding experiments operate on the tags and not on the symbols directly. The last example is than $(t_{j-1}, t_j) = (\text{number}_{\text{inside}}, \text{currency}_{\text{begin}})$ and $(t_{j-1}, t_j) = (\text{number}_{\text{begin}}, \text{currency}_{\text{begin}})$.

Additionally, we evaluated morphological features like part-of-speech tags and lemma features, a *lexical* feature modeling a word within a window of source positions (α to be a set with $\alpha \leq$ some constant), and *and*-combinations of *lexical* features (called *source-n-grams* within the chapter about grapheme-to-phoneme conversion, Chapter 4, and statistical machine translation, Chapter 5). However, these features do not improve over the baselines constructed with the more general features enumerated before.

3.2.2.2 Phrase-Based Translation

Phrase based translation as introduced in Section 1.3.2 is adapted to the task of semantic tagging. The segmentation provided by the corpora is used directly in the generation of the translation model (also known as extraction of phrases), without using an external word aligner such as GIZA++. In parallel, a language model is extracted from the target part of the bilingual corpora, as large monolingual corpora are missing for semantic tagging.

Hypothesis ranking is done with respect to source-to-target and target-to-source phrase models, source-to-target and target-to-source lexical models, the language model, word penalties, and phrase penalties. The log-linear combination of these models is optimized with respect to the concept error rate (CER) (see Section 3.2.5). Reordering parameters are set to nearly zero to avoid reordering, as the semantic tagging tasks does not need a reordering.

3.2.2.3 Generative Weighted Finite State Transducer Approach

The experiments marked with *FST* within in this chapter are implemented by a consecutive composition ($\circ$) of transducers [Mohri 09, Allauzen & Riley$^+$ 07]:

$$\lambda_{SLU} = \text{project}_{tupel}(\lambda_G \circ \lambda_{gen} \circ \lambda_{w2c}) \circ \lambda_{SLM}[\circ\lambda_v]$$

The arcs are weighted with the application of a trigram language model of the joint probability of words and tags in λ_{SLM}

$$p(w_1^J, t_1^J) = \prod_{j=1}^{J} p((w_j, t_j)|(w_{j-2}, t_{j-2}), (w_{j-1}, t_{j-1})) \tag{3.16}$$

The component acceptors/transducers are defined as:

- λ_G a finite acceptor representing the input sequence as a chain, where each state has one arc representing one word except the final state. The arcs are ordered exactly as in the input sequence.
- λ_{gen} is a transducer converting words to predefined classes (e.g. cities, numbers, ...). This may increase generalization for classes of words which are typical unknown words like seldom city names or large numbers.
- λ_{w2c} is a transducer mapping phrases of words (one or many consecutive words) to attribute names. This transducer is weighted and weights are deduced from frequencies calculated with respect to the training data.
- λ_{SLM} is an acceptor modeling a stochastic trigram language model over the word/attribute name tuples (Equation 3.16).
- λ_v is an optional transducer associating one attribute value to a sequence of word/attribute name tuples with the same attribute name.

Here, the experiments are conducted with the AT&T FSM/GRM libraries [Mohri & Pereira$^+$ 02].

3.2.2.4 Support Vector Machines

Within the support vector machine [Vapnik 98] experiments in this chapter, the open source toolkit YAMCHA [Kudo & Matsumoto 01] was applied. It performed best in the CoNLL2000 shared task on chunking and BaseNP chunking [Tjong Kim Sang & Buchholz 00]. As support vector machines do not support sequences directly, it estimates classifiers at every position in a sequence, taking its previous decision as an input feature. YAMCHA estimates these classifiers both in forward and backward direction and combines these classifiers in the final decision. To support multiple classes, the one-versus-one approach is used. The input word sequence is described

by features similar as for the ME model (cf. Section 3.2.2.1), but selected for optimal support vector machine classification. The 1-to-1 alignment approach described in Section 3.2.1 was applied. To make decisions dependent on each other, a bigram feature on the target sequence was applied, similar to maximum entropy Markov models. Support vector machines support overlapping features (as do log-linear models) opening the possibility to use the same features including the lexical and word part features.

3.2.2.5 Dynamic Bayesian Networks

This approach uses dynamic Bayesian networks [Dean & Kanazawa 88] implemented in the graphical model toolkit by [Bilmes & Zweig 02]. The authors define inference over the product of the posterior probability $p(w_1^T|c_1^N)$ and the prior probability $p(c_1^N)$

$$\hat{c}_1^N = \underset{c_1^N}{\operatorname{argmax}}\, p(c_1^N|w_1^T) = \underset{c_1^N}{\operatorname{argmax}}\, p(w_1^T|c_1^N)p(c_1^N) \tag{3.17}$$

The graphical model toolkit supports multiple levels of output labels, leading to the possibility of modeling

$$\hat{c}_1^N, \hat{t}_1^N = \underset{c_1^N, t_1^N}{\operatorname{argmax}}\, p(c_1^N, t_1^N|w_1^T) = \underset{c_1^N, t_1^N}{\operatorname{argmax}}\, p(w_1^T|c_1^N, t_1^N)p(t_1^N|c_1^N)p(c_1^N) \tag{3.18}$$

Probabilities are modeled with factored language models using generalized parallel backoff [Bilmes & Kirchhoff 03] supported by the SRI language model toolkit [Stolcke 02]. Factored language models support dependencies on a vector of stochastic variables. This vector may include the regular surface form of the word as used in standard language models or part-of-speech tags, lemmas, a different variable like w_n in modeling of c_n. Standard language models apply discounting or smoothing techniques like Kneser-Ney discounting [Kneser & Ney 95], but these techniques are not applicable to a vector of stochastic variables. Thus, generalized parallel backoff [Bilmes & Kirchhoff 03] was introduced extending a number of backoff strategies over a graph of backoffs. Alignments may be accounted within the graphical model toolkit as hidden variables and are estimated with the help of EM training.

Within the context of semantic tagging, dynamic Bayesian networks are used to estimate attribute names and attribute values depending on the given word sequence [Lefèvre 07]. The probability is estimated with a set of factored language models [Bilmes & Kirchhoff 03]:
$p(c_1^N) = \prod p(c_i|c_{i-\delta}^{i-1})$: attribute name sequences,
$p(v_1^N|c_1^N) = \prod p(v_i|c_i)$: attribute values conditioned on attribute names,
$p(w_1^T|c_1^N) = \prod p(w_i|w_{i-\delta}^{i-1}, c_i)$: word sequences conditioned on attribute names,
$p(w_1^T|v_1^N, c_1^N) = \prod p(w_i|w_{i-\delta}^{i-1}, v_i, c_i)$: word sequences conditioned on attribute names and values,
with δ the respective n-gram length. In the experiments in this chapter, bigrams and trigrams are used together with the Kneser-Ney smoothing adaptation of generalized parallel backoff. The experiments regarding attribute name extraction do not include models over attribute values. Thus, the models $p(v_1^N|c_1^N)$ and $p(w_1^T|v_1^N, c_1^N)$ are not used in these experiments.

3.2.3 Margin Extension of Conditional Random Fields

The general feature function $H(t, w)$ (for clarity we will drop the positional indices $j = 1, \ldots, J$)

$$H(t, w) = \sum_{r=1}^{R} \lambda_r h_r(t, w) = \boldsymbol{\lambda}\boldsymbol{h}(t, w)$$

can be factorized into a parameter vector $\boldsymbol{\lambda}$ and the feature vector $\boldsymbol{h}(t, w)$. The function H actually can be interpreted as the definition of a hyperplane in the feature space defined by the h_r and

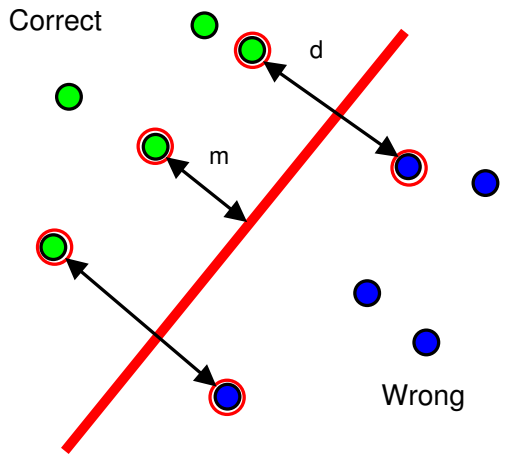


Figure 3.1 Similar to support vector machines large margin training optimizes a hyperplane. However, here the binary classification is not between two given classes but between correct (reference, $\boldsymbol{h}(t_k, w)$) and wrong (all other possible classifications, $\boldsymbol{h}(t, w)$) (Section 3.2.3). The sample points provided as reference in the training set t_k are drawn in green, and the samples $t \neq t_k$ are drawn in blue. Both samples should be separated by a (hyperplane) decision boundary in red. *m* marks the (minimum) distance between the next point to the boundary and the boundary, while *d* marks the (minimum) distance between a correct and a misclassified sample.

defined by the vector $\boldsymbol{\lambda}$ perpendicular to the hyperplane. In Figure 3.1, the examples provided as reference in the training set t_k are drawn in green (class 1), and the examples $t \neq t_k$ are drawn in blue (class -1). Both examples should be separated by a (hyperplane) decision boundary in red defined by $H = 0$. Conditional maximum likelihood maximization (Section 3.2.2.1) optimizes $\boldsymbol{\lambda}$ to maximize the probability $p_{\boldsymbol{\lambda}}(t|w)$ of the training examples. Large margin training instead optimizes $\boldsymbol{\lambda}$ to maximize the margin m between the points next to the boundary and the boundary. Following [Bishop & Nasrabadi 06], p. 327 the size of m is equal to $|H|/\|\boldsymbol{\lambda}\|$. The scale invariance of $\boldsymbol{\lambda}$ permits to define the distance to the nearest point to the boundary (called support vectors) and the boundary to be defined as 1. Thus, every point has to fulfill

$$\tau(t)H(t, w) \geq 1 \tag{3.19}$$

with $\tau(t) = 1$ for correct and $\tau(t) = -1$ for wrong samples. (This equation can only be formulated under the assumption that a hyperplane exist which separates correct and wrong examples. If such a hyperplane does not exist under the chosen structure of $H(t, w)$ a different $H(t, w)$ has to be chosen.) In [Altun & Tsochantaridis$^+$ 03], Section 5, an alternative way of maximization of the margin is given. Instead of the distance between the support vectors and the margin, the distance between a correct and all not correct examples

$$d_{k,t} = H(t_k, w) - H(t, w) \tag{3.20}$$

is expected to be greater than 1:

$$d_{k,t} \geq 1 \qquad \forall t \neq t_k \tag{3.21}$$

Continuing the argumentation of [Bishop & Nasrabadi 06], p. 328, $1/2\|\boldsymbol{\lambda}\|^2$ has to be minimized with respect to the constraint on the distance to the margin. After weakening the constraint in Equation (3.19) to a soft margin $t(t_n)H(t_n, w) \geq 1 - \xi_n$ the objective function can be represented

as ([Bishop & Nasrabadi 06], p. 337)

$$\hat{\boldsymbol{\lambda}} = \underset{\boldsymbol{\lambda}}{\operatorname{argmin}} \left\{ \sum_{k=1}^{K} l(\tau(t_k)t_k) + c||\boldsymbol{\lambda}||^2 \right\} \tag{3.22}$$

with the hinge loss

$$l(w) = \max\{1 - w, 0\} \tag{3.23}$$

Using the distance Equation (3.20), the hinge loss becomes

$$l_{d_k} = l(\min_{t \neq t_k}\{d_{k,t}\}) = \max\{1 - \min_{t \neq t_k}\{d_{k,t}\}, 0\} = \max_{t \neq t_k}\{\max\{1 - d_{k,t}, 0\}\}$$

and the objective function is

$$\hat{\boldsymbol{\lambda}} = \underset{\boldsymbol{\lambda}}{\operatorname{argmin}} \left\{ \sum_{k=1}^{K} l_{d_k} + c||\boldsymbol{\lambda}||^2 \right\} \tag{3.24}$$

the authors of [Zhang & Jin$^+$ 03] proposed to approximate the hinge loss (Equation (3.23)) for binary classification as

$$l_\gamma(w) = \frac{1}{\gamma} \ln\left(1 + \exp\left(-\gamma w\right)\right) \tag{3.25}$$

and proved that $\frac{\ln 2}{\gamma} = \max_w l_\gamma(w) - l(w)$. Together with the distance Equation (3.20) this approximated loss was extended to multiple classes in [Heigold & Schlüter$^+$ 09] to

$$\begin{aligned} l_\gamma(t, d_{k,t}, \rho) &= \frac{1}{\gamma} \ln\left(1 + \sum_{\tilde{t} \neq t} \exp\left(\gamma(-d_{k\tilde{t}} + \rho)\right)\right) && (3.26) \\ &= -\frac{1}{\gamma} \ln\left(\frac{1}{1 + \sum_{\tilde{t} \neq t} \exp\left(\gamma(-d_{k\tilde{t}} + \rho)\right)}\right) \\ &= -\frac{1}{\gamma} \ln\left(\frac{1}{1 + \sum_{\tilde{t} \neq t} \exp\left(\gamma(-H(t,w) + H(\tilde{t},w) + \rho)\right)}\right) \\ &= -\frac{1}{\gamma} \ln\left(\frac{\exp\left(\gamma(H(t,w) - \rho)\right)}{\sum_{\tilde{t}} \exp\left(\gamma(H(\tilde{t},w) - \rho\delta(t,\tilde{t}))\right)}\right) \\ &= -\frac{1}{\gamma} \ln\left(\frac{\exp\left(\gamma(H(t,w) - \rho\delta(t,t))\right)}{\sum_{\tilde{t}} \exp\left(\gamma(H(\tilde{t},w) - \rho\delta(t,\tilde{t}))\right)}\right) && (3.27) \end{aligned}$$

Using the loss Equation (3.27) in the objective function Equation (3.24) results in

$$\begin{aligned} \hat{\boldsymbol{\lambda}} &= \underset{\boldsymbol{\lambda}}{\operatorname{argmin}} \left\{ \sum_{k=1}^{K} -\frac{1}{\gamma} \ln\left(\frac{\exp\left(\gamma(H(t_k,w_k) - \rho\delta(t_k,t_k))\right)}{\sum_{\tilde{t}} \exp\left(\gamma(H(\tilde{t},w_k) - \rho\delta(t_k,\tilde{t}))\right)}\right) + c||\boldsymbol{\lambda}||^2 \right\} \\ &= \underset{\boldsymbol{\lambda}}{\operatorname{argmax}} \left\{ \sum_{k=1}^{K} \frac{1}{\gamma} \ln\left(\frac{\exp\left(\gamma(H(t_k,w_k) - \rho\delta(t_k,t_k))\right)}{\sum_{\tilde{t}} \exp\left(\gamma(H(\tilde{t},w_k) - \rho\delta(t_k,\tilde{t}))\right)}\right) - c||\boldsymbol{\lambda}||^2 \right\} \end{aligned}$$

To support efficient calculation on sequences, $\delta(t_k, \tilde{t})$ was replaced by the word accuracy between the hypothesis $\tilde{t}_1^N$ and the reference $t_{1,k}^N$ $\mathcal{A}(\tilde{t}_1^N, t_{1,k}^N) = \sum_{n=1}^{N} \delta(\tilde{t}_n, t_{n,k})$ reaching the final modified-MMI or modified-CRF proposed in [Heigold & Schlüter$^+$ 09]:

$$\hat{\boldsymbol{\lambda}} = \underset{\boldsymbol{\lambda}}{\operatorname{argmax}} \left\{ \sum_{k=1}^{K} \frac{1}{\gamma} \ln\left(\frac{\exp\left(\gamma(H(t_{n,k}, t_{n-\delta,k}^{n-1}, w_{1,k}^N) - \rho\mathcal{A}(t_{1,k}^N, t_{1,k}^N))\right)}{\sum_{\tilde{t}} \exp\left(\gamma(H(\tilde{t}_n, \tilde{t}_{n-\delta}^{n-1}, w_{1,k}^N) - \rho\mathcal{A}(t_{1,k}^N, \tilde{t}_1^N))\right)}\right) - c||\boldsymbol{\lambda}||^2 \right\} \tag{3.28}$$

Note that only the training of the $\boldsymbol{\lambda}$ and not the decision process is changed. The decision process is still the maximization of H with respect to the input vector w_1^N (Equation (3.11)).

The Modified-CRF formulation has some favorable properties:

- With $\gamma = 1$ and $\rho = 0$ the conditional log likelihood optimization Equation (3.8) is preserved.
- With $\gamma \to \infty$ and $\rho = 1$ we have a support vector machine.

With the help of the meta-parameters c, γ, ρ it is possible to scale between regular conditional random fields and support vector machines. The maximum in Equation (3.28) is invariant with respect to a scaling factor γ. c controls the size of $||\boldsymbol{\lambda}||^2$ and $\boldsymbol{\lambda}$ the size of all H. As a consequence, only the balance between the two summands H and $\rho\mathcal{A}$ is important, and this balance can already be tuned with the regularization constant c. In the experiments, we used $\gamma = \rho = 1$.

Expanding the logarithms in Equation 3.28 and using $\mathcal{A}(t_{1,k}^N, t_{1,k}^N) = 1$ results in the equivalent representation:

$$\hat{\boldsymbol{\lambda}} = \underset{\boldsymbol{\lambda}}{\operatorname{argmax}} \left\{ \sum_{k=1}^{K} \left\{ H(t_{n,k}, t_{n-\delta,k}^{n-1}, w_{1,k}^N) - \ln \sum_{\tilde{t}} \exp\left(H(\tilde{t}_n, \tilde{t}_{n-\delta}^{n-1}, w_{1,k}^N) + \frac{\rho}{\gamma}(1 - \mathcal{A}(t_{1,k}^N, \tilde{t}_1^N)) \right) - c||\boldsymbol{\lambda}||^2 \right\} \right\}$$

With $\gamma = \rho = 1$ this is equivalent to the Softmax-Margin in [Gimpel & Smith 10] (Equation (6) in this publication). There, the objective function is minimized resulting in an inverted sign and the cost is used instead of an accuracy. The cost is there a Hamming distance as used here.

3.2.4 Implementation of Conditional Random Fields

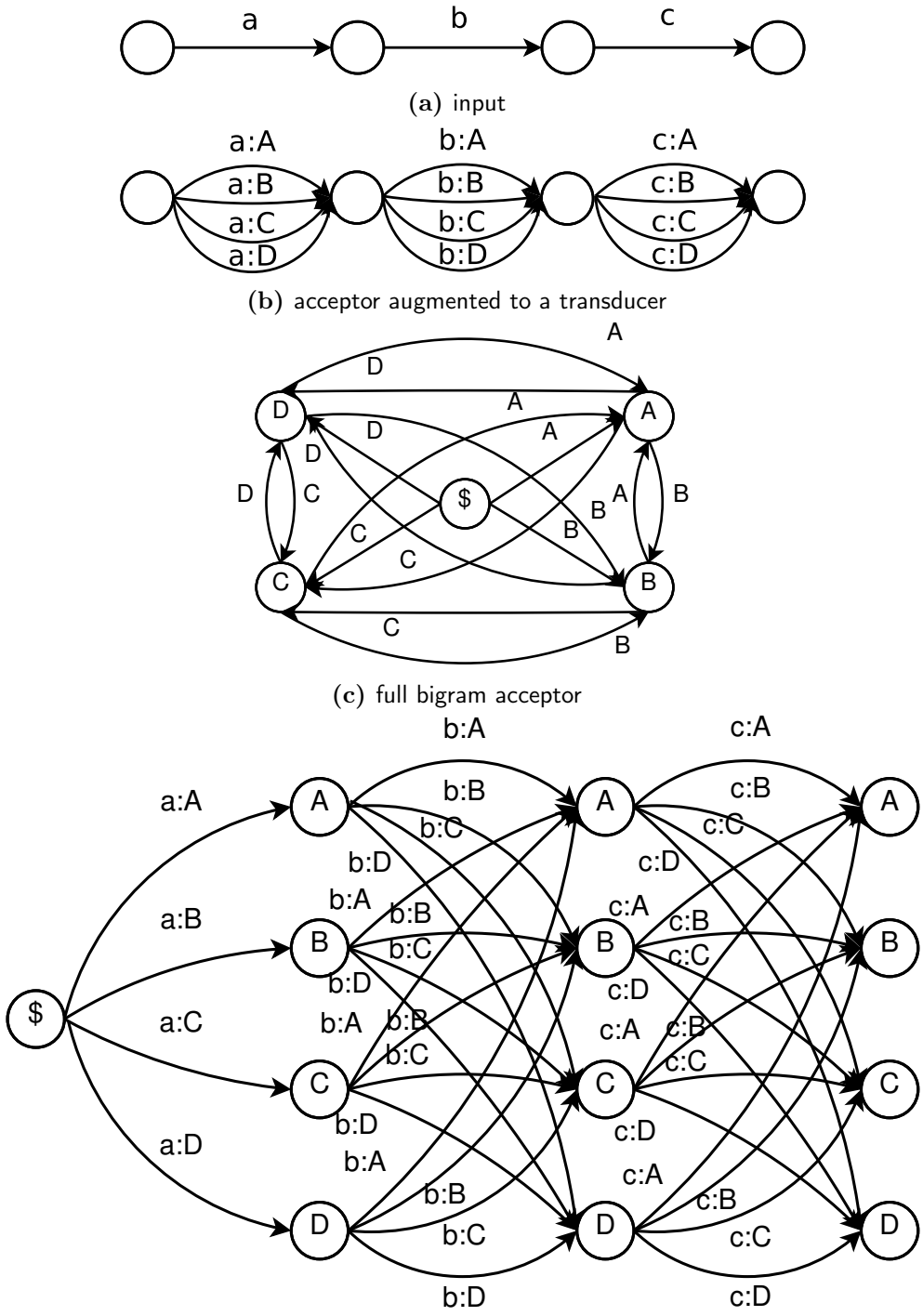
Initially, we used the open source implementation CRF++ [Kudo 05] to model, estimate and apply conditional random fields (Section 3.2.2.1) experiencing already high accuracies. At this point, there was already a maximum mutual information implementation within the in-house automatic speech recognition. Maximum mutual information is the same training criterion as in conditional log-likelihood within conditional random fields. Thus, we decided to use this code base to reimplement conditional random fields based on finite state transducers [Mohri 09, Allauzen & Riley$^+$ 07]. Figure 3.2 visualizes the construction of such a transducer based on a simplified input vocabulary a, b, c and output vocabulary A, B, C, D. Only the finite state transducer operations *augmentation* and *composition* with a fully connected bigram acceptor are needed. In the final transducer, at each arc, the input symbol w_j, the output symbol t_j, and the history of the output symbol t_{j-1} is known, enabling the application of the general feature function $H(t_{j-1}, t_j, w_j)$. With a forward and backward operation on the transducer, all needed input symbols can be selected from w_1^J enabling the application of

$$H(t_{j-1}, t_j, w_1^J) = \sum_{r=1}^{R} \lambda_r h_r(t_{j-1}, t_j, w_1^J)$$

resulting in the final weighted transducer Figure 3.3. Within the estimation, the derivatives

$$\begin{aligned} \frac{\partial L}{\partial \lambda_r} &= \sum_{k=1}^{K} \sum_{j=1}^{J} \left(h_r(t_{j-1}, t_j, w_1^J) - \sum_{\tilde{t}_{j-1}, \tilde{t}_j} p(\tilde{t}_{j-1}, \tilde{t}_j, w_1^J) h_r(\tilde{t}_{j-1}, \tilde{t}_j, w_1^J) \right) \\ &= N_r - D_r \end{aligned}$$

need to be calculated, which can be accomplished by the application of the finite state transducer posterior operation with a logarithmic semiring $(+, -log(\exp(-a) + \exp(-b)))$ and a traversal of each arc in the resulting transducer. The traversal is implemented as a depth first search



(a) input

(b) acceptor augmented to a transducer

(c) full bigram acceptor

(d) final topology of transducer

Figure 3.2 Build-up of the conditional random fields within a finite state transducer framework. The example uses the input vocabulary a, b, c and the output vocabulary A, B, C, D. The input sequence "a b c" is represented as a chain acceptor with one arc per input symbol (a). Each arc of this acceptor is duplicated for each output vocabulary symbol resulting in a transducer (b), and composed with a fully connected bigram acceptor (c). In the final transducer (d) at each arc the previously applied output symbol, the current output symbol, and the input symbol are known.

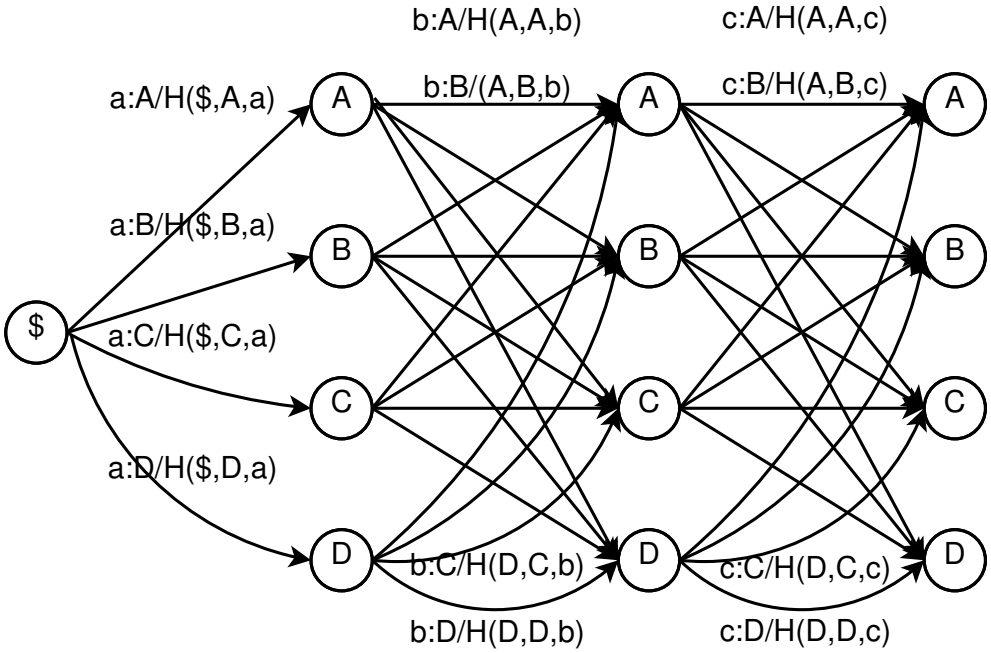


Figure 3.3 Scores with CRF features $H(t_{j-1}, t_j, w_1^J) = \sum_{r=1}^{R} \lambda_r h_r(t_{j-1}, t_j, w_1^J)$ applied to transducer in Figure 3.2d. For simplicity w_1^J is only represented with the word w_j aligned with t_j. Only a subset of the arcs have been labeled to keep the figure readable. Notation of the labels is "input symbol":"output symbol"/"weight".

accumulating the statistics D_r as a sum of the products $p(\tilde{t}_{j-1}, \tilde{t}_j, w_1^J) h_r(\tilde{t}_{j-1}, \tilde{t}_j, w_1^J)$ and the statistics N_r by accumulating $h_r(t_{j-1}, t_j, w_1^J)$ at every arc which is allowed with respect to the reference. Search can be conducted with the help of a single source shortest distance (SSSD) operation with respect to a tropical semiring $(+, \max\{a, b\})$.

$$\underset{t_1^J}{\operatorname{argmax}} \left\{ p(t_1^J | w_1^J) \right\} = \underset{t_1^J}{\operatorname{argmax}} \left\{ \prod_{j=1}^{J} \exp\left(H(t_{j-1}, t_j, w_1^J) \right) \right\} = \underset{t_1^J}{\operatorname{argmax}} \left\{ \sum_{j=1}^{J} H(t_{j-1}, t_j, w_1^J) \right\}$$

One advantage of the finite state transducer implementation is that the search space may be represented in a human readable way through processing with a drawing software as *dot*. Additionally, constraints on transitions can be easily implemented. E.g. within the semantic tagging experiments, sequences as $\mathrm{A}_{\mathrm{begin}} \rightarrow \mathrm{B}_{\mathrm{inside}}$ may be prohibited. However, we gained higher accuracies by permitting these transitions and finally replacing such a transition $\mathrm{A}_{\mathrm{begin}} \rightarrow \mathrm{B}_{\mathrm{inside}}$ with $\mathrm{A}_{\mathrm{begin}} \rightarrow \mathrm{B}_{\mathrm{begin}}$.

The approach described here is modified in chapter 4 for higher computational efficiency. Mainly, the weights are applied as early as possible and the fully connected bigram acceptor is replaced with a partially connected n-gram acceptor with backing off.

3.2.5 Experimental Results for Attribute Name Extraction

Three objectives raise from the theoretical discussion:

1. *Which features are useful in the context of semantic tagging with respect to conditional random fields?*
2. *Does the margin extension of conditional random fields give a tagging performance gain?*
3. *How do the methods compare with respect to tagging performance gain?*

Table 3.2 Feature build-up of the CRF system on the French MEDIA corpus, including the number of active features.

	features	number of features	CER [%] TRAIN	DEV	EVAL
1	(t_j, w_j)	419 900	82.6	91.6	88.9
2	+ (t_{j-1}, t_j)	456 190	9.7	15.3	14.6
3	+ $(t_j, w_{j-1}) + (t_j, w_{j+1})$	1 247 730	3.9	13.1	12.4
4	+ capitalization	1 247 920	3.8	13.0	12.0
5	+ prefixes	1 683 210	3.5	12.8	11.5
6	+ margin-posterior	**1 683 210**	**10.0**	**12.3**	**10.6**
7	$(t_j, w_{j-1}) + (t_j, w_j) + (t_j, w_{j+1})$	1 230 761	15.4	25.5	24.7
8	$(t_j, w_{j-1}) + (t_j, w_{j+1}) + (t_{j-1}, t_j)$	840 901	15.5	25.1	22.9

Table 3.3 Feature build-up of the CRF system on the Polish LUNA corpus, including the number of active features.

	features	number of features	CER [%] TRAIN	DEV	EVAL
	$(t_j, w_{j-1}) + (t_j, w_j) + (t_j, w_{j+1})$				
1	+ (t_{j-1}, t_j)	4 550 728	1.1	25.7	26.1
2	+ prefixes	5 725 352	1.1	22.8	23.5
3	+ suffixes	6 982 320	1.1	22.0	22.7
4	+ capitalization	6 982 696	1.1	21.8	22.6
5	+ margin-posterior	**6 982 696**	**6.9**	**21.0**	**21.5**

Tagging performance is evaluated with respect to the *concept error rate* (CER). The concept error rate is defined as the sum of the Levenshtein distances between the reference annotations provided with the corpus and the hypotheses generated with some method from the source sequences provided by the corpus, divided by the sum of the lengths of references in the corpus. The Levenshtein distance is defined by the number of insert, delete, and substitution edit operations needed to obtain the reference from the hypothesis. Prior to scoring, the *NULL* tag representing out of domain groups is removed from hypothesis and reference. The concept error rate was evaluated by adopting the NIST scoring toolkit for word error rates in automatic speech recognition [NIST 95]. The corpora are described in detail in Section 3.1. Experiments are reported with respect to three corpora of recorded telephone calls: the French MEDIA corpus from the tourist domain, the Polish LUNA corpus composed of calls to the Warsaw transportation system, and the Italian LUNA corpus including calls to a computer help desk.

3.2.5.1 Which features are useful in the context of semantic tagging with respect to conditional random fields?

In Section 3.2.2, lexical features, (word part) prefix features, (word part) suffix features, a capitalization feature, and target bigram features are introduced. Additionally, we evaluate in early experiments morphological part-of-speech tag features, lemma features, lexical in a window (bag-of-words), and source-n-grams. Lexical in a window (bag-of-words) and source-n-grams do not improve performance, while morphological part-of-speech tag features and lemma features do not improve a system with (word part) prefix features and (word part) suffix features. The second set is much simpler to use as it does not need an external parser. Thus, we decided to use only word part features.

Table 3.4 Feature build-up of the CRF system on the Italian LUNA corpus, including the number of active features.

	features	number of features	CER [%] TRAIN	DEV	EVAL
	$(t_j, w_{j-3}) + \ldots + (t_j, w_{j+1})$				
1	$+ (t_{j-1}, t_j)$	882 414	3.5	24.7	24.4
2	+ prefixes	1 163 970	2.7	22.4	20.6
3	+ suffixes	1 475 010	2.6	22.6	20.1
4	+ margin-posterior	**1 475 010**	**17.7**	**20.6**	**20.0**

Table 3.5 Features used with conditional random fields on the various corpora ("cap." denotes the capitalization feature).

corpus	lexical	bigram	cap.	prefix	suffix	# features
French	$w_{j-1}, \ldots, w_{j+1}$	✓	✓	1 . . . 4	-	1 683 210
Polish	$w_{j-1}, \ldots, w_{j+1}$	✓	✓	1 . . . 4	1 . . . 4	6 982 696
Italian	$w_{j-3}, \ldots, w_{j+1}$	✓	-	1 . . . 6	1 . . . 6	1 475 010

For all systems we first optimized the regularization on a mininum set of features ($(t_j, w_{j-1}) + (t_j, w_j) + (t_j, w_{j+1}) + (t_{j-1}, t_j)$) on the development set (DEV), followed by a more accurate selection of the lexical features, than prefixes, suffixes, capitalization, and finally the regularization parameter is tuned again. Actually, the regularization parameter never needed to be changed in the re-tuning. Table 3.2 documents a feature build up on the French MEDIA corpus. The most important features are the lexical feature for the current word (t_j, w_j) and the target bigram feature (t_{j-1}, t_j) resulting in half a million features (all features have been used including features not seen in the references of the training corpus) and a tagging performance of 15.3% CER on the development corpus. The next significant improvement of 15% relative is gained by the neighboring lexical features $(t_j, w_{j-1}) + (t_j, w_{j+1})$, while (word part) prefix features together with a capitalization feature only give a small final improvement of 2% relative.

Line 7 and 8 in Table 3.2 justify that bigram features (t_{j-1}, t_j) and the current word feature (t_j, w_j) are crucial for the final performance of the system. Both systems are a variation of the system in line 3. The system in line 7 lacks the bigram features (t_{j-1}, t_j) and the system in line 8 lacks the current word feature (t_j, w_j). Leaving them out gave a more than 90% relative regression in CER.

The optimal size of the lexical features window (in MEDIA $-1, \ldots, 1$), the optimal length of the (word part) prefix and suffix features, and whether or not a capitalization feature was useful, is dependent on the choice of the corpus. Table 3.5 reports the optimal choice of features with respect to the selected corpus and Table 3.3 and Table 3.4 show the respective build up on the Polish LUNA and the Italian LUNA corpora. However, on all corpora mainly a lexical window of $-1, \ldots, 1$ and target bigram already give a competitive tagging performance.

The features that we call (word part) prefix and suffix features are not linguistically correct morphological prefixes or suffixes: They are only the first or last letters of a lexical unit. Thus, the prefixes of a word might with high probability include the word stem. This might justify why a word part prefix may be useful in a French text. Additionally, the word part features have increased the stability of the system to incorrect spelling of the word. As only the prefix and/or the suffix needed to be correct, a lexical unit, which might be confused with a similar lexical unit by the automatic speech recognizer, can still be tagged correctly.

3.2.5.2 Does the margin extension of conditional random fields give a tagging performance gain?

As reported in the last lines of Tables 3.2, 3.3, and 3.4, the margin extension improves tagging performance with respect to CER on most sets of the the three corpora. On the development sets of the MEDIA corpus and the Polish LUNA corpus an improvement of 0.5%/0.8% absolute and 3%/4% relative, while on the evaluation part of these corpora an improvement of 1.0% absolute and 8%/5% relative was reached. On the Italian LUNA corpus the figure is not consistent. While the tagging performance increased strongly compared to the other corpora by 2.0% absolute and 9% relative on the development set, the tagging performance does practically not change on the evaluation set. An interesting change can be seen in the tagging performance on the training sets. With the use of the margin extension it is drastically increased and reaches on the French MEDIA corpus and the Italian LUNA corpus a similar performance as on the development and evaluation set. This behavior is in general desirable, as the generalization was increased from the training to the test sets and a common problem of maximum entropy models is alleviated.

3.2.5.3 How do the methods compare with respect to tagging performance gain?

In the LUNA project we were able to compare the maximum entropy models in this thesis with respect to a set of different models. The models were tested in two different setups. In the *text input* setup the models process the human transcribed input of the recorded telephone calls and in the *speech input* setup the models are applied on calls processed with automatic speech recognition. It can be expected that there are much more errors in the automatically processed input. The recognition performance of the used speech recognizer is documented in the caption of Table 3.6 and was in the range of 30%-40% WER. The comparison of models is documented in Table 3.6.

On all corpora, conditional random fields models with the margin extension are the best choice by far in tagging performance for reference input of the French MEDIA corpus and for the automatic speech recognizer input on all corpora. In the setup using reference input for the Italian and Polish LUNA corpora finite state transducers reached similar results. However, these results do not generalize to noisy speech input.

Maximum entropy Markov models reach only average performance with significant worse tagging performance compared to conditional random fields. The only difference between both models are the normalization, which is performed at the position level for maximum entropy Markov models and at sentence level for conditional random fields. An analysis at the tag level (Table 3.7) has shown that the context dependent commands *+/-command-** are significantly better tagged with conditional random fields.

3.3 Design of an Attribute Value Extraction Module within the LUNA Context

The task of concept tagging can be seen as a hierarchical or multi-layer categorization. First, the sequence of words is segmented and associated to an attribute name, the general category of a concept. Second, the words aligned to an attribute name are normalized to an attribute value. E.g., as in the example of Figure 3.4, the word *une* (one) is categorized to the attribute name *nb_chambre* (number of hotel rooms) and normalized to the value *1.* Or, more general, the MEDIA corpus defines three types of attribute values:

numeric units processed for example with regular expressions

proper names extracted only with basic normalization, e.g. spaces

semantic class e.g. "comparative" with possible values "around", "less-than", "maximum", "minimum" and "more-than"

Table 3.6 Results of attribute name extraction on French MEDIA, Polish and Italian LUNA. System results (CER [%]) on the manually (text input) and automatically (speech input) transcribed DEV and EVAL corpora. Results are ordered by CER on EVAL on speech input. The WER for speech input for French is 30.3% on DEV and 31.4% on EVAL, for Polish 39.5% on DEV and 38.9% on EVAL and for Italian 28.5% on DEV and 27.0% on EVAL.
[1]provided by Marco Dinarelli, University of Trento, now CNRS-LaTTiCe
[2]provided by Christian Raymond, University of Avignon, now Univ. IRISA-INSA
[3]provided by Fabrice Lefèvre, Université d'Avignon et des Pays de Vaucluse

		CER [%]			
		text input		speech input	
		a. name		a. name	
	model	DEV	EVAL	DEV	EVAL
French	Conditional Random Fields (CRFs)	12.3	10.6	24.0	23.8
	Support Vector Machines (SVMs)[1]	14.2	13.4	27.1	25.8
	Maximum Entropy Markov Models (MEMMs)	15.8	13.7	26.6	26.4
	Finite State Transducers (FSTs)[1,2]	16.1	14.1	28.3	27.5
	Phrase Based Translation (PBT)	16.0	15.1	28.4	29.0
	Dynamic Bayesian Networks (DBNs)[3]	17.0	15.5	29.5	29.1
Polish	Conditional Random Fields (CRFs)	21.0	21.5	53.6	51.7
	Maximum Entropy Markov Models (MEMMs)	24.0	25.1	58.0	57.0
	Dynamic Bayesian Networks (DBNs)[3]	27.5	26.6	58.9	57.7
	Finite State Transducers (FSTs)[1,2]	20.5	21.9	58.3	57.9
	Support Vector Machines (SVMs)[1]	26.2	27.3	59.1	58.1
	Phrase Based Translation (PBT)	27.2	27.7	60.3	59.0
Italian	Conditional Random Fields (CRFs)	20.6	20.0	30.0	28.4
	Dynamic Bayesian Networks (DBNs)[3]	24.3	25.7	33.6	32.1
	Maximum Entropy Markov Models (MEMMs)	24.6	27.3	33.2	33.3
	Finite State Transducers (FSTs)[1,2]	22.1	20.1	35.6	33.3
	Phrase Based Translation (PBT)	25.0	25.0	35.0	33.7
	Support Vector Machines (SVMs)[1]	24.6	25.3	36.3	34.0

Table 3.7 Comparing tagging of maximum entropy Markov models (MEMM) and conditional random fields (CRF) based on per word tags on the French MEDIA corpus. The begin and continue tags are merged into the categories *null*, *command* (attribute names: *+command-dial*, *+command-tache -command-tache*) and *other* (begin and continue tags of every attribute name excluding *null* and *command*). All results in % CER.

	total	MEMM		CRF		improvement
null	9534	881	(9.2%)	821	(8.6%)	7%
command	2160	705	(32.6%)	465	(21.5%)	34%
other	14982	2719	(18.2%)	2320	(15.5%)	15%
Sum	26676	4305	(16.2%)	3606	(13.5%)	16%

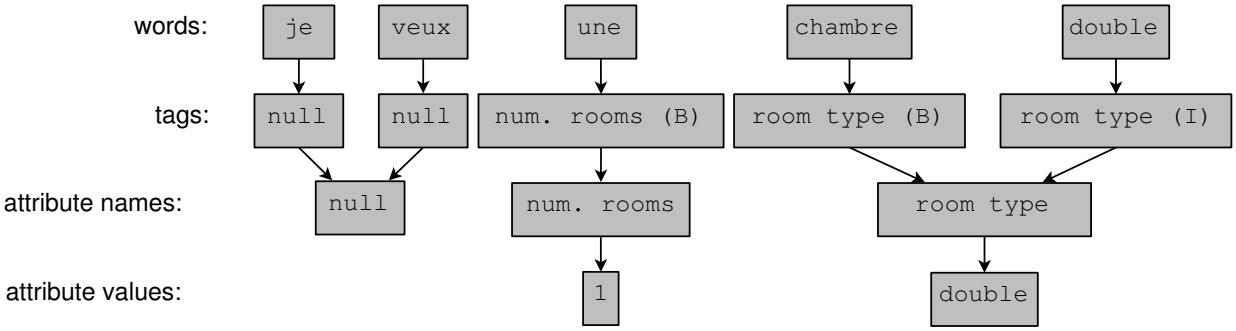


Figure 3.4 Illustrating the use of attribute values with respect to attribute names. (French: "I would like a double-bed room"). The first line shows the input word sequence, the third and fourth line the appropriate attribute names and values (original names in corpus: *nombre-chambre* and *chambre-type*). The second line shows how the 1-to-1 alignment is modeled using "begin" (B) and "inside" (I) tags.

In this section, we compare two different approaches to extract the attribute value given the attribute name and the words aligned to this attribute name. The first approach is a programmatic deterministic approach. A Perl script and later in another setup a set of un-weighted finite state transducers are used to process the words aligned to an attribute name. E.g. based on the attribute name *room type* a Perl function or a finite state transducer is selected to convert the word sequence *chambre double* to *double*. This approach works very well when a clear conversion rule can be formulated. Examples are numbers or categories which can be exactly defined with lists of examples. However, such deterministic approaches get in-efficient when there is no clear rule or the spoken language has a large variance. Examples are confirmation responses which are often more complex than exact *yes/no* answers. Responses like *hmm*, *yeah*, *I don't think so* are common as well. For such contexts we investigated statistical attribute value extraction. We based the recognition of attribute values $c^I_{1,v}$ with given attribute names $c^I_{1,n}$ and w^J_1 on the probability $p(c^I_{1,v}|c^I_{1,n}, w^J_1)$. We modelled this probability with conditional random fields and similar features as in the attribute name extraction.

In first experiments we applied either the rule based or the statistical approach. However, it turned out that performance of both approaches changed based on the respective attribute name. For some attribute names the rule based and for some the statistical approach was best. So a combination heuristic was implemented.

3.3.1 Rule Based Approach

Hand-crafted rules can be both modeled as finite state transducers or in a procedural way with a script. For each type of input (numbers, choices, ...) a finite state transducer or a script function can be designed mapping the input words to attribute values, and these transducers or methods can be combined to process the respective attribute name. Finite state transducers have the advantage that an n-best or lattice of possible input words may be processed [Raymond & Béchet$^+$ 06, Servan & Raymond$^+$ 06]. However, as only first best automatic speech recognizer input is used, the script is sufficient and is used in the experiments.

3.3.2 Constrained Conditional Random Fields

Supporting a second output layer on top of a first output layer is subject to some constraints. First, the alignment between first and second output layer is exactly one-to-one. The approach should generate exactly the same number of symbols in the first and second output layer. Second, there

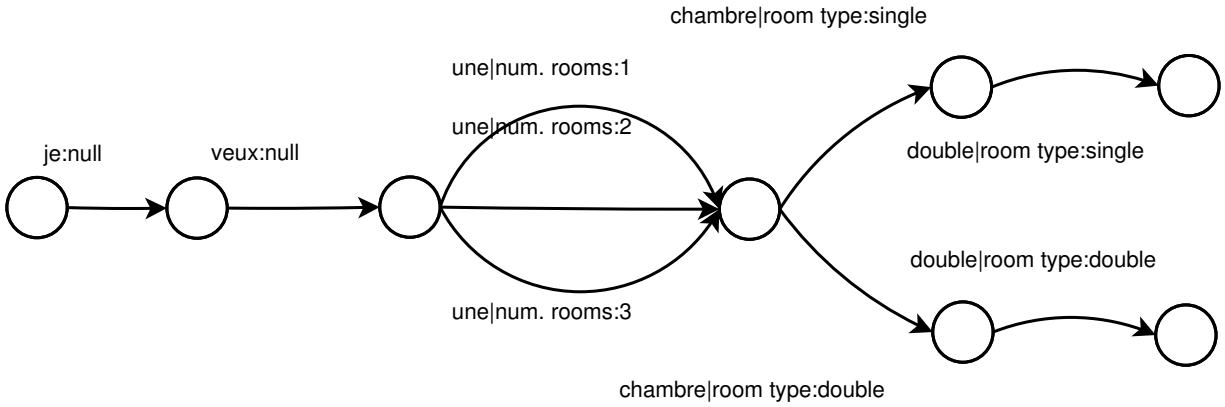


Figure 3.5 Possible search graph of constrained conditional random fields of an attribute value extraction. The sequence *je veux une chambre double* (French: *I would like one double room*) is already tagged with attribute names "num. rooms" and "room type" (original names in corpus: *nombre-chambre* and *chambre-type*) the search adds the possible attribute values after the colon. The correct attribute value extraction is *je veux {une}|{num. rooms}:1 {chambre double}|{room type}*:double.

should be the possibility to switch off labeling of some attribute names. E.g., in some cases, numbers may introduce an unnecessary complexity, as they may more easily be processed by a deterministic method. Third, for each attribute name, there may be distinct sets of possible attribute values. E.g., the approach should not generate a number if a *yes/no* decision is needed. Additionally, without that restriction, the possible set of output labels becomes huge and computation time not acceptable any more. However, in some cases the sets of possible attribute values may be reused for different attribute names. E.g., numbers may be bus numbers, house numbers, number of hotel rooms, etc.

We took the approach described in Section 3.2.4 and Figure 3.3 and extended it. The recognition is modelled with finite state transducers. Finite state transducers have at each arc an input symbol, an output symbol, and a weight. However, here we need to model three symbols per arc. The aligned word, the attribute name, and the attribute value. We have chosen to save the aligned word with the attribute name in the input symbol separated by | characters. The attribute value is the output symbol. From the training data we extracted all attribute values aligned to an attribute name. E.g. for the attribute name *room type* only the two values *single* and *double* had been extracted. The output symbols were restricted to this map. Per attribute name the recognition was additionally restricted to only produce one attribute value. All arcs permitting a change in the attribute name are removed. If the attribute value extraction is switched off for an attribute name, only a *null* label is used. An example is given in Figure 3.5.

3.3.3 Combination of Conditional Random Fields and the Rule Based Approach

As already described in the beginning of the section, both methods can be combined based on the attribute names. Both methods propose an attribute value per attribute name, including possibly the *null* tag with constrained conditional random fields. Thus, it can be decided globally which method is best for which attribute name. E.g., a probabilistic method may have low performance with sparse attribute values like bus stop names. At the same time it may be more stable with *yes/no* decisions as it is able to model probability from context. The selection of the method for each attribute name is determined by performance of each method with respect to a specific attribute name on the development set. Errors are counted for each attribute name and the best

Table 3.8 Comparison of rule-based and statistical attribute value extraction and their combination for the conditional random fields approach on all tasks covered in this chapter (CER[%]).

		CER [%]					
		reference input		text input		speech input	
	extraction method	DEV	EVAL	DEV	EVAL	DEV	EVAL
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

method is chosen.

3.3.4 Attribute Value Extraction in Dynamic Bayesian Networks

With the choice of models for dynamic Bayesian networks the posterior probability of the input words is conditioned on both the attribute names and attribute values. However, a full search space with all combinations of attribute names and attribute values is not feasible. Thus, the attribute values are first marginalized and the attribute names are hypothesized first. In a second step the attribute values are extracted using the full probability and respecting the selected attribute names.

3.3.5 Experimental Results for Attribute Value Extraction

Table 3.8 compares the rule-based approach, the statistical approach, and their combination. Performance is measured with respect to the concept error rate (CER, see Section 3.2.5), with the difference that a concept is defined as the combination of an attribute name with an attribute value. Whereas in Section 3.2.5 the concept was just an attribute name. For each setup (reference, text input, speech input) the selection for the best method per attribute name was chosen on the respective attribute name extraction on the development set. Reference setup is the choice of attribute names given in the reference. The result in this setup is given for completeness, but cannot be used to present final results as the use of the reference can be considered as cheating. In the reference setup, the rule-based approach is every time superior to the statistical approach. However, the margin melds down as the input gets noisy and even vanishes on speech input of the French MEDIA and the Italian LUNA corpus. And, in all cases except reference input and speech input of the Italian LUNA corpus, the combination of both approaches has higher performance than each method separately.

Finally, Table 3.9 summarizes all results including the results solely on attribute names and result of attribute names with attribute values in both, the text and speech input. Taking attribute value extraction into account, conditional random fields as designed within this thesis are the best performing method on all corpora and both setups even only by using the rule-based attribute value extraction. Using additionally the combination of both approaches (numbers in braces) increases the margin by one percentage point in CER absolute. The increased margin in performance with respect to sole attribute name extraction suggests that not only the attribute name sequence is extracted with high performance, but additionally the segmentation of the input has a high accuracy.

3.4 Conclusion

Within this chapter, an attribute name and attribute value extraction module using conditional random fields and maximum entropy Markov models have been designed. The final conclusion is that conditional random fields have the highest performance with respect to concept error rate. The margin in concept error rate to other models even increases with noisier input (speech input) and a tuple of output labels (attribute name with attribute value). Sophisticated features (word-part, capitalization) and the margin extension to conditional random fields do help but are not critical to a good performing system.

The reason why conditional random fields have that high accuracies is still an open question. If a guess is permitted, we would guess that conditional random fields benefit from the high quality references and they model labels in a better way which depend on context information. In [Rubenstein & Hastie$^+$ 97], models were trained in an artificial setup to compare generative and discriminative training with the conclusion: "It is best to use an informative approach if confidence in the model correctness is high." (here "informative approach" is the generative approach). A slight misfit in the model is penalized less in a discriminative approach opposed to a generative approach. Perhaps by more accurate design of the generatively trained models, the difference to conditional random fields would vanish. However, conditional random fields are a good starting point giving already high performance without a sophisticated feature/model design.

The scientific contributions of this chapter have already been published in the journal article [Hahn & Dinarelli$^+$ 11] and the conference proceedings [Lehnen & Hahn$^+$ 09, Hahn & Lehnen$^+$ 09, Heigold & Lehnen$^+$ 08, Hahn & Lehnen$^+$ 08a, Hahn & Lehnen$^+$ 08b].

The work presented in this chapter was the work of multiple authors. The contributions of the different authors can be summarized as follows:

- Which features are useful in the context of semantic tagging with respect to conditional random fields?
  The CRF/MEMM software used in all experiments presented in this thesis was mainly written by *Lehnen*. The work was based on the MMI training in the ASR modules originally written by *Heigold*. *Hahn* helped at various places. Feature functions, preparation of the experimental setup, the sparse parameter implementation. *Guta* and *Riess* implemented feature functions.
- Does the margin extension of conditional random fields give a tagging performance gain?
  The idea and implementation came from *Heigold*. *Hahn* and *Lehnen* helped with conducting and analysing experiments.
- How do the methods compare with respect to tagging performance gain?
  Design of experiments: The idea for the experiment came from *Hahn*. Experiments and data pre- and post-processing was done by *Hahn* and *Lehnen* together. The CRF and MEMM experiments were done by *Hahn*, *Lehnen*, *Guta*, *Riess*, and *Heigold*. The other methods shown in comparison were prepared by PBT: *Hahn*, *Vilar*; SVM: *Dinarelli*[1], FST: *Dinarelli*[1], *Raymond*[2] , DBN: Lefevre[3].
- Constrained conditional random fields (attribute values)
  The idea and implementation came from *Lehnen*.
- Rule based approach (attribute values)
  This approach was developed and the results were provided by *Dinarelli*[1] and *Raymond*[2].

- Combination of constrained conditional random fields and the rule based approach (attribute values)
  Idea, implementation and experiments by *Lehnen* and *Hahn*.
- Attribute value extraction in dynamic Bayesian networks
  This approach was developed and the results were provided by *Lefevre*[3].

If not stated otherwise the authors were part of the RWTH Aachen University during the mentioned research.

[1] Marco Dinarelli, University of Trento, now CNRS-LaTTiCe
[2] Christian Raymond, University of Avignon, now Univ. IRISA-INSA
[3] Fabrice Lefèvre, Université d'Avignon et des Pays de Vaucluse

Table 3.9 Extension of Table 3.6 including attribute values on top of attribute names. A combination of attribute name and value is assumed to be correct only if both labels are correct with respect to the reference in columns headed with *a. name & value*. Results in concept error rate (CER [%]) are presented (CER [%]) the manually (text input) and automatically (speech input) transcribed DEV and EVAL corpora. The WER for speech input for French is 30.3% on DEV and 31.4% on EVAL, for Polish 39.5% on DEV and 38.9% on EVAL and for Italian 28.5% on DEV and 27.0% on EVAL. *stat. a. value* refers to a combination of statistical and rule-based attribute value extraction used for the CRF approach. The results are ordered by the last column. All other figures use the same rule-based approach.
[1]provided by Marco Dinarelli, University of Trento, now CNRS-LaTTiCe
[2]provided by Christian Raymond, University of Avignon, now Univ. IRISA-INSA
[3]provided by Fabrice Lefèvre, Université d'Avignon et des Pays de Vaucluse

		CER [%]							
		text input				speech input			
		a. name		a. name & value		a. name		a. name & value	
	model	DEV	EVAL	DEV	EVAL	DEV	EVAL	DEV	EVAL
French	CRF	12.3	10.6	15.2	13.5	24.0	23.8	29.0	28.2
	+ stat. a. value			14.5	12.6			28.6	**27.3**
	SVM[1]	14.2	13.4	17.2	15.9	27.1	25.8	31.5	29.7
	MEMM	15.8	13.7	18.2	16.3	26.6	26.4	31.4	30.7
	FST[1,2]	16.1	14.1	18.3	16.6	28.3	27.5	32.5	31.3
	DBN[3]	17.0	15.5	19.3	17.4	29.5	29.1	34.6	32.8
	PBT	16.0	15.1	18.8	17.8	28.4	29.0	33.3	33.5
Polish	CRF	21.0	21.5	26.4	26.3	53.6	**51.7**	59.7	57.3
	+ stat. a. value			24.5	24.7			59.1	**56.7**
	SVM[1]	26.2	27.3	30.3	31.2	59.1	58.1	63.3	61.5
	MEMM	24.0	25.1	29.1	30.0	58.0	57.0	63.1	61.7
	DBN[3]	27.5	26.6	33.2	31.4	58.9	57.7	64.8	63.1
	FST[1,2]	20.5	21.9	26.1	27.1	58.3	57.9	65.3	64.0
	PBT	27.2	27.7	33.6	33.6	60.3	59.0	66.2	64.4
Italian	CRF	20.6	20.0	22.2	22.4	30.0	28.4	33.1	32.1
	+ stat. a. value			21.7	21.8			32.5	**31.3**
	DBN[3]	24.3	25.7	26.2	28.9	33.6	32.1	37.2	36.3
	SVM[1]	24.6	25.3	25.8	27.1	36.3	34.0	39.7	36.7
	MEMM	24.6	27.3	26.3	29.3	33.2	33.3	36.9	37.0
	FST[1,2]	22.1	20.1	24.2	23.1	35.6	33.3	39.4	37.2
	PBT	25.0	25.0	27.4	27.9	35.0	33.7	38.8	37.5

4. Applying Maximum Entropy Approaches to Grapheme-to-Phoneme Conversion

Conditional random fields have lead to large improvements in tasks with given or simple 1-to-1 alignments and a limited number of features as part-of-speech (POS) tagging [Lafferty & McCallum$^+$ 01], semantic tagging (chapter 3), or chunking [Sha & Pereira 03]. However, taking the transition to more complex tasks, as statistical machine translation, introduces complex feature sets, huge vocabularies, and a non-monotonic many-to-many alignment problem. Grapheme-to-phoneme conversion is an excellent task for investigations on scaling conditional random fields to monotonic many-to-many alignments using huge amount of complex features, with the advantage of exploring a method with a high potential in conversion performance for grapheme-to-phoneme conversion. In this chapter we present a general many-to-many alignment extension to conditional random fields and reduce computational time and memory consumption via elastic-net with RPROP, sparse-forward-backward, and beam pruning in training. The final system reaches state-of-the-art performance on the English pronunciation dictionaries Celex and Pronlex.

4.1 Introduction

We evaluate our hidden conditional random fields implementation on the task of grapheme-to-phoneme (G2P) conversion. Even as the vocabularies are small in grapheme-to-phoneme conversion, it includes already two important challenges: Monotonous alignments are needed and the models need to use complex features taking into account multiple source symbols together with multiple target symbols.

The contributions of this chapter are

- an efficient HCRF implementation adopting the *BIO scheme* [Ramshaw & Marcus 95] providing a word alignment with overlapping phrase features,
- an extension of the RPROP algorithm to support elastic-nets,
- sparse-forward-backward for arbitrary n-gram sizes,
- beam pruning in HCRF training,
- analysis of the effect of joint-n-grams,
- a training scheme avoiding external alignments and external search space constraints.

Parts of this chapter have already been published in the conference proceedings [Lehnen & Hahn$^+$ 11a, Hahn & Lehnen$^+$ 11, Lehnen & Hahn$^+$ 11b]. This chapter integrates all presented approaches in one framework, evaluates them additionally on the grapheme-to-phoneme conversion task Pronlex, and adds beam-pruning and joint-n-gram features. The small deviations in the reported error rates compared to the conference proceedings can be explained by a corrected error in feature count cut-off.

An overview of the related work in the field of grapheme-to-phoneme conversion has already been given in Section 1.2.2. This chapter will continue with a recapitulation of our basic implementation,

Table 4.1 Statistics of the English CELEX and PRONLEX pronunciation dictionaries.

	# symbols		avg. word length		# unique words		
	source	target	source	target	TRAIN	DEV	EVAL
CELEX	26	58	8.4	7.1	39 985	5 000	15 000
PRONLEX	30	41	7.4	6.9	83 182	2 400	4 800

and the corpora used for evaluation (Section 4.3). This basic system will then be extended in two directions: the integration of a hidden alignment (Section 4.4) and with improvements with respect to time and memory consumption (Section 4.6). Finally, all extensions will be evaluated together in Section 4.7. Additionally, a comparison to a conditional random fields system designed for phrase based translation is included in Section 4.8.

4.2 Corpora

Grapheme-to-Phoneme conversion is the task of mapping letter sequences to phoneme sequences, e.g. "phoenix" to "finIks". In contrast to the natural language processing corpora a segmentation between letters and phonemes is generally not shipped with the corpora and has to be generated as part of the grapheme-to-phoneme approach. In chapter 4, experiments are reported conducted on the two English corpora CELEX and PRONLEX (statistics in Table 4.1).

The English CELEX corpus consists of 60k randomly selected pronunciations from the original English CELEX database [Baayen & Piepenbrock$^+$ 95]. It is based on a mixture of British and American text sources, finally including both pronunciation styles. The source vocabulary are the 26 Latin letters used in the English vocabulary and the target vocabulary is a set of 58 phonemes represented in the SAMPA notation. The same partitioning of the corpus as in [Bisani & Ney 08] is used.

The second corpus is an American pronunciation dictionary, PRONLEX. The training set has approximately double the size of the CELEX corpus, with a bit smaller phoneme set. Thus, a doubling in training time could be expected for most methods including conditional random fields. Additionally to the 26 latin letters, the source lexicon permits the additional symbols dot, hyphen, underscore and single quote. The complete corpus statistics for both corpora are given in Table 4.1.

4.3 Core System

In the course of this chapter, the conditional random fields [Lafferty & McCallum$^+$ 01] software already used in the last chapter for attribute name and attribute value extraction is extended from linear chain conditional random fields (LCCRFs, Equation 3.7) to hidden conditional random fields (Section 4.4), which is done by adopting the implementation described in Section 3.2.4.

Similar to the parameterization in Section 3.2.2.1, we use in this chapter a linear combination of binary feature $h_r(y_{i-\delta}^{i-1}, y_i, x_1^J) \in \{0, 1\}$ for the general feature description

$$H(y_{i-\delta}^{i}, x_1^J) = \sum_{r=1}^{R} \lambda_r h_r(y_{i-\delta}^{i-1}, y_i, x_1^J). \tag{4.1}$$

As a consequence, $H(y_{i-\delta}^{i}, x_1^J)$ is a sum over the parameters λ_r of all active ($h_r(\ldots) = 1$) binary features. The features are active if a condition on their parameters $y_{i-\delta}^{i-1}, y_i, x_1^J$ is fulfilled. We mostly follow here the proposal of [Jiampojamarn & Cherry$^+$ 10], which reports top level results on CELEX. [Jiampojamarn & Cherry$^+$ 10] uses only surface features taking only combinations

of multiple letters on source side and phonemes on the target side into account. We define the features as

- **single**: source to target features depending on one source symbol with the current target symbol $(y_n, x_{A(i)+m})$ with $m = -5 \dots 5$, $A(i) = i$ for linear chain conditional random fields and the aligned word in hidden conditional random fields (Section 4.4),
- **source n-grams**: and-combinations of the *single* features $(y_i, x_{A(i)-\gamma_1}^{A(i)+\gamma_2})$ with $\gamma_1, \gamma_2 = 0, \dots, 5$, $\gamma_1 + \gamma_2 + 1 \leq 6$.
- **target n-grams**: target transition features $(y_{i-\delta}^{i-1}, y_i)$ with some length δ, and
- **joint n-grams**: combinations of *source n-grams* and *target n-grams*.

In the final setups, only these feature functions were actually used which have been encountered two times for *joint n-grams* and one time for all other features in the training references (feature count cut-off of 1). This largely reduced the number of needed derivatives of Equation 4.5 ($< 1\%$) and did not hurt the performance.

4.4 Hidden Conditional Random Fields (HCRFs)

In chapter 3, the correspondence of output symbols to input symbols is normally provided with the training corpus and can be modeled as a 1-to-1 alignment (see Section 3.2.1). If an output symbol can be assigned to more than one input symbol, the alignment can be modeled as a 1-to-1 alignment with the help of the *BIO scheme* [Ramshaw & Marcus 95]. In the *BIO scheme* the output symbols are augmented with *begin*, *inside*, and *outside* labels to obtain a 1-to-1 alignment. Without a given alignment in the training corpus, two strategies are possible: (A) An alignment is generated with an external method, which models the probability of an alignment A, given the source sequence $x_1^J = x_1, \dots, x_J$ and the target sequence $y_1^I = y_1, \dots, y_I$: $p(A|x_1^J, y_1^I)$. But in search the target is not known, resulting in a different probability $p(y_1^I, A|x_1^J)$ or $p(y_1^I|x_1^J) = \sum_A p(y_1^I, A|x_1^J)$. Additionally, an external method has the disadvantage of error propagation into the final method. Or, (B) by direct integration of the alignment as a hidden variable in the model. This provides the possibility to train specialized alignment features and training all features with the knowledge of the hidden alignment. In [Quattoni & Wang$^+$ 07, Koo & Collins 05, Gunawardana & Mahajan$^+$ 05] CRFs summing over a predefined graph capturing the hidden structure are introduced and named hidden conditional random fields (HCRFs). The graph has to be constrained in some way, e.g. in [Koo & Collins 05] by a part-of-a speech parse tree. In [Yu & Lam 08], the use of a predefined graph was mainly avoided; only the number of hidden states per seen state was limited to keep the training feasible showing improvements on (POS) tagging and a named entity recognition task. In this chapter, we will show how to use the idea of the *BIO scheme* [Ramshaw & Marcus 95] to implement an efficient HCRF trainer for arbitrary monotonic alignments. To generate more than one output symbol per input symbol, we will adapt the idea of HMMs to a monotonous translation task. One adaptation is that the skip arcs in HMMs are replaced by arcs adding two target symbols.

Linear chain conditional random fields (LCCRFs), introduced in [Lafferty & McCallum$^+$ 01], model the conditional probability of a target sequence of length I $y_1^I, y \in \mathbb{Y}$ and, if needed, an alignment A with respect to a source sequence $x_1^J, x \in \mathbb{X}$ with the help of the features $h_r(y_{i-\delta}^i, A, x_1^J)$ and parameters λ_r ($r \in \{1, \dots, R\}$).

$$p(y_1^I, A|x_1^J) \overset{LCCRF}{=} \frac{\exp\left(\sum_{i=1}^{I} \sum_{r=1}^{R} \lambda_r h_r(y_{i-\delta}^i, A, x_1^J)\right)}{\sum_{\tilde{A} \in \mathbb{A}} \sum_{\tilde{y}_1^I} \exp\left(\sum_{i=1}^{I} \sum_{r=1}^{R} \lambda_r h_r(\tilde{y}_{i-\delta}^i, \tilde{A}, x_1^J)\right)}. \tag{4.2}$$

In our systems, the feature functions are binary features $h_r(y_{i-\delta}^i, A, x_1^J) \in \{0, 1\}$ resulting in very sparse feature vectors h_1^R permitting to use efficient calculations of the product with the parameters

λ_1^R.

The alignment A is needed to cover mappings where one or more source symbols are mapped to one or more target symbols. A grapheme-to-phoneme conversion example would be the two letters `p h` mapped to the single phoneme `f` in Figure 1.3 (M-to-1 alignment), or the single letter `x` mapped to the two phonemes `k s` in Figure 1.3 (1-to-N alignment). Covering all possible alignments A is in general infeasible and rarely needed. With grapheme-to-phoneme conversion only monotonous alignments with small shifts between source and target are needed (see Section 1.2.2). We implemented the alignment restriction $\mathbb{A}$ implicitly by adopting the *BIO scheme* [Ramshaw & Marcus 95] in Section 4.5. As in many tasks, grapheme-to-phoneme conversion corpora do rarely provide a reference alignment. Here, a summation with respect to the *hidden* alignment is a possible solution $p(y_1^I|x_1^J) = \sum_{A\in\mathbb{A}} p(y_1^I, A|x_1^J)$ leading to:

$$p(y_1^I|x_1^J) \overset{HCRF}{=} \frac{\sum_{A\in\mathbb{A}} \exp\left(\sum_{i=1}^{I}\sum_{r=1}^{R} \lambda_r h_r(y_{i-\delta}^i, A, x_1^J)\right)}{\sum_{\tilde{A}\in\mathbb{A}}\sum_{\tilde{y}_1^I} \exp\left(\sum_{i=1}^{I}\sum_{r=1}^{R} \lambda_r h_r(\tilde{y}_{i-\delta}^i, \tilde{A}, x_1^J)\right)} \tag{4.3}$$

The integration of Equation (4.3) is called hidden conditional random fields (HCRFs, [Quattoni & Wang$^+$ 07, Koo & Collins 05, Yu & Lam 08]). It has to be noted that by integrating a hidden variable into conditional random fields, the hidden conditional random fields training is no longer convex and there is a risk of optimizing to an optimum which is not the global optimum. In Section 4.5.3, linear chain conditional random fields with an externally derived alignment are contrasted with hidden conditional random fields with the conclusion that hidden conditional random fields have a lower error rate.

The regularized parameters λ_1^R are estimated via maximization of the conditional log-likelihood averaging over $n = 1, \ldots, N$ training samples $\ldots, (\overline{y}_{1,n}^I, x_{1,n}^J), \ldots$ with

$$\begin{aligned} L_{LCCRF} &= \sum_{n=1}^{N} \log\left(p\right)\left(\overline{y}_{1,n}^I, \overline{A}_n | x_{1,n}^J\right) - \sum_{\kappa=1}^{2} c_\kappa ||\lambda_1^L||_\kappa^\kappa \\ L_{HCRF} &= \sum_{n=1}^{N} \log\left(p\right)\left(\overline{y}_{1,n}^I | x_{1,n}^J\right) - \sum_{\kappa=1}^{2} c_\kappa ||\lambda_1^L||_\kappa^\kappa \,, \end{aligned} \tag{4.4}$$

and a reference alignment $\overline{A}_n$ in the case of linear chain conditional random fields. Parameters λ_1^L are found by calculating the partial derivatives

$$\begin{aligned} \frac{\partial L}{\partial \lambda_r} &= \sum_{n=1}^{N} N_{r,n} - \sum_{n=1}^{N} D_{r,n} - c_1 \frac{\partial|\lambda_r|}{\partial \lambda_r} - c_2\lambda_r \\ N_{r,n} &= \sum_{i=1}^{I} h_r(\overline{y}_{i-\delta,n}^i, \overline{A}_n, x_{1,n}^J) \quad \text{(LCCRF)} \\ N_{r,n} &= \sum_{A\in\mathbb{A}} p(A|x_{1,n}^J) \sum_{i=1}^{I} h_r(\overline{y}_{i-\delta,n}^i, A, x_{1,n}^J) \quad \text{(HCRF)} \\ D_{r,n} &= \sum_{\tilde{A}\in\mathbb{A}} \sum_{y_1^I} p(y_1^I|x_{1,n}^J) \sum_{i=1}^{I} h_r(y_{i-\delta}^i, \tilde{A}, x_{1,n}^J) \end{aligned} \tag{4.5}$$

and estimating $\frac{\partial L}{\partial \lambda_r} = 0$. $p(A|x_1^J)$ is Equation (4.2) without the sum over y_1^I in the denominator. $N_{r,n}$ is the expectation value of the function h_r with respect to the training data and $D_{r,n}$ is the expectation value of h_r with respect to the estimated model. The optimization of this function will be described in detail in Section 4.6.1.

As a consequence of the Bayes decision rule with 0/1-loss, the best target sequence in search is found by maximizing the conditional probability for a source sequence x_1^J:

$$\hat{y}_1^I = \underset{\hat{y}_1^I}{\operatorname{argmax}} \left\{ p(y_1^I | x_1^J) \right\} = \underset{\hat{y}_1^I}{\operatorname{argmax}} \left\{ \frac{\sum_{A \in \mathbb{A}} \exp\left(\sum_{i=1}^{I} \sum_{r=1}^{R} \lambda_r h_r(y_{i-\delta}^i, A, x_1^J) \right)}{\sum_{\tilde{A} \in \mathbb{A}} \sum_{\tilde{y}_1^I} \exp\left(\sum_{i=1}^{I} \sum_{r=1}^{R} \lambda_r h_r(\tilde{y}_{i-\delta}^i, \tilde{A}, x_1^J) \right)} \right\}$$

The denominator is constant and thus does not change the argument of the maximization. Experiments have shown that most of the probability mass is already allocated to the first best alignment. A sum in the alignment does not change the recognition result. So the sum in the alignment can be replaced by a maximization resulting in the simpler Equation 4.6.

$$\hat{y}_1^I = \underset{\hat{y}_1^I, \hat{A}}{\operatorname{argmax}} \left\{ \sum_{i=1}^{I} \sum_{r=1}^{R} \lambda_r h_r(\hat{y}_{i-\delta}^i, \hat{A}, x_1^J) \right\} \tag{4.6}$$

In the first of the subsequent subsections, the extension from 1-to-1 to M-to-1 alignments is described, which is going to be extended in the second subsection to M-to-N alignments.

4.5 Implementation of the Alignment

Grapheme-to-phoneme conversion needs mappings of single letters to multiple phonemes (`x` $\mapsto$ `k s`) or multiple letters to one phoneme (`p h` $\mapsto$ `f`). Thus, linear chain conditional random fields (Equation (4.2)) and hidden conditional random fields (Equation (4.3)) make use of an alignment variable A. As the summands of the numerator statistics $N_{r,n}$ are a small subset of the summands of the denominator statistics $D_{r,n}$, most of the computation time of (hidden) conditional random field training is spent in the calculation of denominator statistics $D_{r,n}$ (Equation (4.5)). *Without* alignment (1-to-1 mapping, $I = J$), this calculation is implemented for linear output with dynamic programming

$$D_{r,n} = \sum_{i=1}^{I} \sum_{y_{i-\delta}^i} p(y_{i-\delta}^i | x_{1,n}^I) h_r(y_{i-\delta}^i, x_{1,n}^I) \tag{4.7}$$

with (repetition of Equations 3.10)

$$p(y_{i-\delta}^i | x_1^I) = \frac{\alpha_i(y_{i-\delta}^{i-1} | x_1^I) \exp\left(\sum_{r=1}^{R} \lambda_r h_r(y_{i-\delta}^i, x_1^I) \right) \beta_{i+1}(y_{i-\delta}^i | x_1^I)}{Z}$$

$$\alpha_i(y_{i-\delta}^{i-1} | x_1^I) = \sum_{y_{i-2}} \exp\left(\sum_{r=1}^{R} \lambda_r h_r(y_{i-1-\delta}^{i-1}, x_1^I) \right) \alpha_{i-1}(y_{i-\delta-1}^{i-2} | x_1^I)$$

$$\beta_{i+1}(y_{i-\delta}^i | x_1^I) = \sum_{c_{i+1}} \exp\left(\sum_{r=1}^{R} \lambda_r h_r(y_{i+1-\delta}^{i+1}, x_1^I) \right) \beta_{i+2}(y_{i-\delta+1}^{i+1} | x_1^I)$$

$$\beta_{N+1} = 1, \quad \alpha_1 = 1, \quad Z = \beta_0(\$).$$

We will adopt the *BIO scheme* [Ramshaw & Marcus 95] in order to implement efficient training with dynamic programming. In the first of the subsequent subsections, the extension from 1-to-1 to M-to-1 alignments is described, and extended to M-to-N alignments in the second subsection. Using graphones as in the generative baseline from [Bisani & Ney 08] would end in many alignments representing the same hypothesis, which would increase the number of summands drastically. This is acceptable if only a sum over all alignments is carried out as in [Bisani & Ney 08] but prohibitive in combination with a sum over target-n-grams ($\sum_{y_{i-\delta}^i}$).

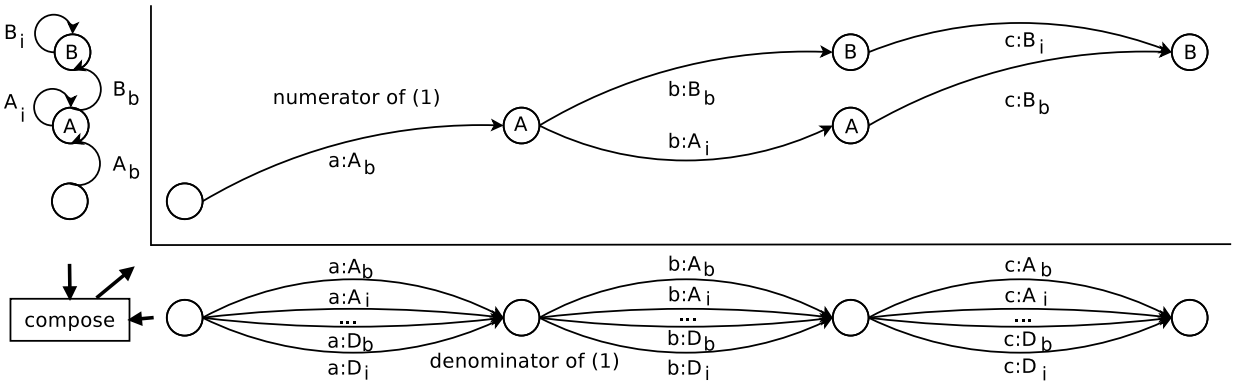


Figure 4.1 Extension of conditional random fields from Section 3.2.4 and Figure 3.2 with M-to-1 alignments (see Section 4.5.1). At the bottom the target symbols of the automaton in Figure b has been augmented with begin markers *b* and continuation markers *i*. This automaton is composed with a 0-1 hidden Markov model resulting in all possible segmentations of the reference with respect to M-to-1 alignments.

4.5.1 M-to-1 Alignment

To support alignments of M source symbols to one target symbol, we propose to use a new vocabulary $\mathbb{T} = \{y\_t : y \in \mathbb{Y}, t \in \{b, i\}\}$. A reverse mapping $Y(t_1^J)$ interprets each symbol y_b as the beginning of a new target symbol y and y_i as the continuation of the last target symbol (map to ϵ). Similarly, a reverse mapping for $A(t_1^J)$ is given. Now, the sum over all hypotheses y_1^I and all acceptable alignments $\tilde{A} \in \mathbb{A}$ can be combined in one sum guided by the source sequence position $j = 1, \ldots, J$ with an adapted δ:

$$
\begin{aligned}
N_{r,n} &\overset{HCRF}{=} \sum_{j=1}^{J} \sum_{t_{j-\delta}^{j} : Y(t_1^J) = \overline{y}_{1,n}^{I}} p(A(t_1^J)|x_{1,n}^J) h_r(t_{j-\delta}^{j}, x_{1,n}^J) \\
D_{r,n} &= \sum_{j=1}^{J} \sum_{t_{j-\delta}^{j}} p(t_{j-1}, t_j|x_{1,n}^J) h_r(t_{j-\delta}^{j}, x_{1,n}^J)
\end{aligned}
\tag{4.8}
$$

Figure 4.1 describes the implementation of the M-to-1 alignment in the finite state transducer framework (Section 3.2.4 and Figure 3.2). For simplicity, only the augmented automaton from Figure b is extended at the bottom of Figure 4.1, but the software actually uses *target n-grams* and in the final configuration *joint n-grams* and in consequence more complicated automatons. At the bottom of Figure 4.1, the target symbols {"A","B","C","D"} are augmented by the begin marker "_b" and the continuation marker "_i" doubling the number of arcs. Now, all possible segmentations $\tilde{A}$ with M-to-1 alignments of any sequence y_1^I are already encoded in the automaton providing the denominator of Equation (4.3) and the statistics $D_{r,n}$ in Equation (4.8). To calculate the $N_{r,n}$ statistics in Equation (4.8), the reference y_1^I needs to be selected with every possible segmentation with M-to-1 alignments. This is implemented by composition with an automaton similar to a 0-1 hidden Markov model (HMM) (on the left in Figure 4.1), resulting in the automaton in the center of Figure 4.1. To calculate $p(A(t_1^J)|x_{1,n}^J)$, the same posterior algorithm can be used as for $p(t_{j-1}, t_j|x_{1,n}^J)$ with respect to the composition result instead of the full transducer.

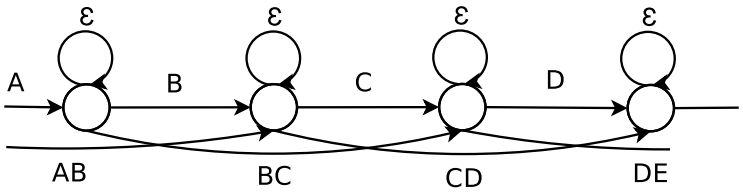


Figure 4.2 Hidden Markov model with skip nodes replaced by arcs emitting two words.

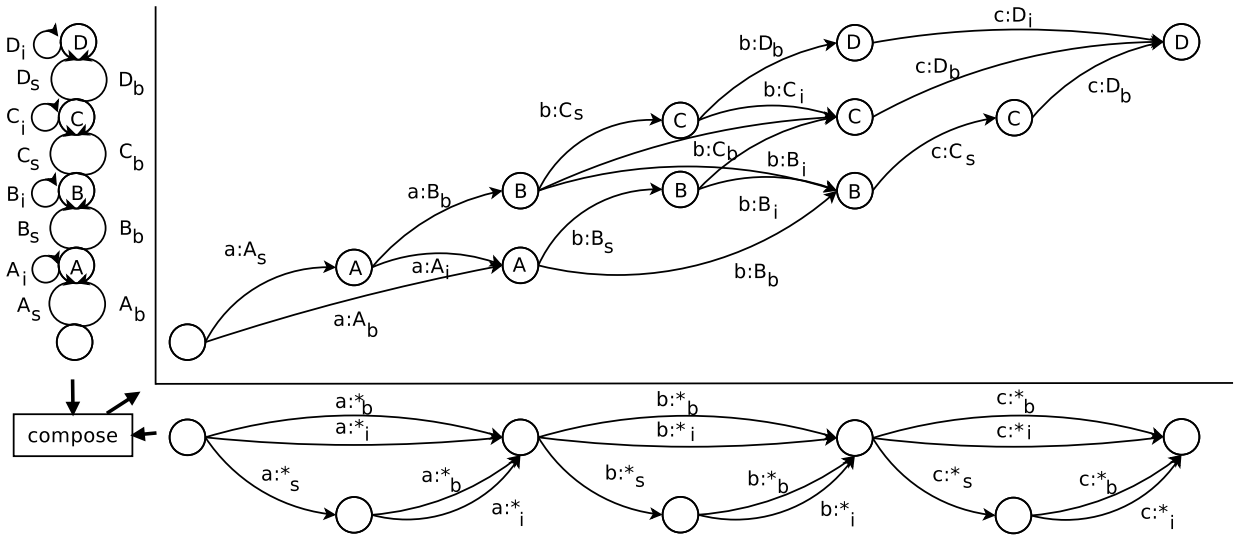


Figure 4.3 M-to-N extension of conditional random fields (see Section 4.5.2). Additionally to the regular path in Figure 4.1, a second path to every arc in Figure b is introduced. On this path, two instead of one possible target symbols can be added, which is marked by the index *s*. To select all possible reference segmentations with respect to M-to-N alignments, the automaton on the left is composed with the automaton at the bottom resulting in segmentations similar to 0-1-2 hidden Markov models in the center.

Table 4.2 Influence of the transition "_s"→"_i" in Figure 4.3. Without this transition exactly a discriminative 0-1-2 HMM is build. With this transition the decision of introducing an additional target symbol is independent from the beginning and continuation decision (Section 4.5.3). Results are reported with respect to the CELEX corpus.

extract numerator	DEV PER[%]	DEV WER[%]	EVAL PER[%]	EVAL WER[%]
0-1-2 HMM (without "_s"→"_i")	2.8	14.1	2.8	13.7
independent decisions (with "_s"→"_i")	2.6	12.6	2.5	12.3

4.5.2 M-to-N Alignment

The implementation to hidden conditional random fields until this point supports the mapping of M source symbols to one target symbol (e.g. the two letters `p h` mapped to the single phoneme `f` in Figure 1.3), but it does not support the mapping of one or many letters to many phonemes (e.g. the single letter `x` mapped to the two phonemes `k s` in Figure 1.3). At this point, it is similar to 0-1 hidden Markov models, either beginning or continuing a target symbol. The longer 0-1-2 hidden Markov models model additionally the skipping of a target symbol. While in automatic speech recognition a phoneme is begun, continued, or not spoken, in grapheme-to-phoneme conversion every phoneme has to be used. Thus, the skip is replaced by producing two instead of one target symbols (Figure 4.2). This is implemented by adding an alternative path to the regular paths in the automaton shown at the bottom of Figure 4.1. The set of possible tags $\mathbb{T}$ is extended by skip tags $y\_s, y \in \mathbb{Y}$. This tags are used on the alternative path to mark a new target symbol. This symbol can be either continued "_i" or a second target symbol can be emitted "_b". The result is demonstrated at the bottom of Figure 4.3. By permitting a skip marker "_s" followed by an inside marker "_i", the transducer in the bottom of Figure 4.3 can be replaced by a consecutive decision. First y'_s or an ϵ, followed by y_b or y_i. Without the path "_s"→"_i", three paths as in 0-1-2 HMMs are possible either generating no new symbol, one new symbol, or two new symbols. In Table 4.2, it is reported that the independent decisions result in a higher performance. From a mathematical point of view the introduction of the alternative path can be described as n-grams on target $\tilde{t}$, which are $(y\_i), (y\_b), (y'\_s, y\_b), (y'\_s, y\_i)$. Now, t in Equation (4.8) is replaced by $\tilde{t}$, and Y is extended to support $\tilde{t}_1^J$. The feature functions are activated two times within the alternative path. One time for the skip-tag and a second time in the begin-/inside-tag. The search Equation (4.6) is now

$$\hat{y}_1^I = Y\left(\underset{\tilde{t}_1^J}{\operatorname{argmax}}\left\{\sum_{j=1}^{J}\sum_{r=1}^{R}\lambda_r h_r(\hat{t}_{j-\delta}^{j}, x_1^J)\right\}\right) . \tag{4.9}$$

With the automaton at the bottom of Figure 4.3, the denominator of Equation (4.3) and the statistics $D_{r,n}$ in Equation (4.5) can be estimated. All reference segmentations are found by composition with the automaton on the left in Figure 4.3 resulting in segmentations similar to 0-1-2 hidden Markov models. This method models all monotonic alignments assigning no new target symbol, one new target symbol, or two new target symbols per source symbol. Permitting three or four target symbols per source symbol is possible with additional doubling arcs (_s). Note that with this alignment modulation, for each source symbol, all aligned target symbols are given and for each target symbol all aligned source symbols are given. Thus, the full word based alignment matrix A is given. Other than the baseline approaches, the phrase modulation is left to the features and not encoded in the search graph.

With 0-1-2 hidden Markov models, usually three penalties are introduced: forward δ_1, loop δ_0, and skip δ_2. These penalties can be tuned empirically on the dev set applying the downhill simplex

Table 4.3 Experiments contrasting 0-1-2 penalties estimation as *subsequent* optimization on the development set or *integrated* within the conditional random fields training to an experiment without penalties (Section 4.5.3). Results are reported wrt. the CELEX corpus.

HMM weights tuning	DEV		EVAL	
	PER[%]	WER[%]	PER[%]	WER[%]
none ($\delta_j = 0$)	25.1	85.6	25.0	85.5
subsequent	3.1	15.3	3.1	15.5
integrated	2.6	12.6	2.5	12.3

algorithm [Nelder & Mead 65] after the conditional random fields model training, or trained as regular features within the conditional random fields framework. Table 4.3 shows that the penalties are needed to gain state-of-the-art results and that penalties implemented as conditional random fields features perform best.

4.5.3 Experimental Results for Hidden Conditional Random Fields

Within the discriminative baseline from [Jiampojamarn & Cherry$^+$ 10], M-to-N alignments are implemented by a preprocessing step generating alignments A on the training references with an external tool. To simulate this strategy, the approach presented in [Bisani & Ney 08] is used to generate an alignment on training data and linear chain conditional random fields are trained on this data. In search, again the approach from [Bisani & Ney 08] is used to double source symbols needed to permit M-to-N alignments, and the trained linear chain conditional random fields are used to generate target symbols. Hidden conditional random fields are trained with exactly the same features (and penalties δ_0, δ_1, δ_2) and compared to the linear chain conditional random fields with external alignment (cf. Table 4.4). Following [Parihar & Picone 02], the improvement in WER for target-2-grams (line 1 to 5) is significant wrt. a significance level of 6% and the improvement for target-3-grams (line 2 to 6) is significant wrt. a significance level of 9% only. However, the improvement in WER is consistent over both sets and both target-n-gram configurations with an average of 4.0% relative improvement and 0.6% of standard deviation. To reduce the probability to get stuck in a poor local optimum, the *single* features corresponding to the current word are initialized with IBM-1 probabilities obtained by GIZA++ [Och & Ney 03]. Line 3 in Table 4.4 is included to show that the traditional configuration of a discriminative 0-1-2 HMM with the current source symbol, target-bigram and $\delta_0, \delta_1, \delta_2$ weights is not sufficient on this task.

4.6 Scaling (Hidden) Conditional Random Fields

Taking into account larger vocabularies or complex features with millions of instances, the original computation of CRFs becomes prohibitively time and memory consuming, already in the domain of grapheme-to-phoneme conversion. On the one hand, the computation time of CRFs is dominated by a polynomial complexity. The degree of the polynomial is equal to the size of the largest features with respect to their span on output symbols. Furthermore, CRFs can process overlapping features, permitting dozens of features to be active at the same time and millions of features in total. A common strategy is again to use an external method to restrict the possible hypotheses in training and search. In automatic speech recognition, it is common to use word lattices generated by a strong generative baseline [Woodland & Povey 02], while in statistical machine translation, the authors of [He & Deng 12] have claimed to get improvements using 100-best lists. However, with the use of an external method, the final integration of the CRF approach and the generative methods remains tricky. Both have to use the same vocabularies, exactly the same preprocessing and the

Table 4.4 Comparison of linear chain conditional random fields (LCCRFs) and hidden conditional random fields (HCRFs) trained with the same features (plus penalties δ_j with hidden conditional random fields), showing improvements with hidden conditional random fields on the CELEX corpus. Line 3 is the traditional configuration of a 0-1-2 HMM. The linear chain conditional random fields make use of an external segmentation extracted from recognition results gained with [Bisani & Ney 08]. (Section 4.5.3)

		DEV		EVAL	
		PER[%]	WER[%]	PER[%]	WER[%]
	LCCRFs with external segmentation				
1	target-2-grams (same features as 5)	2.7	13.1	2.6	12.9
2	+ target 3-grams (same features as 6)	2.7	12.7	2.5	12.1
3	HCRF with features: single + (δ_j)	52.5	97.1	52.7	97.7
4	+ source n-grams	4.0	20.9	3.8	20.2
5	+ target-2-grams	2.6	12.6	2.5	12.3
6	+ target-3-grams	2.6	12.3	2.5	11.6

hypotheses where the CRF approach can result in improvements have to be in the word lattices or n-best lists. To reduce the complexity of the CRF estimation directly, the authors of [Lavergne & Cappé$^+$ 10] have proposed to use elastic-net (EN) (a combination of L1- and L2-regularization, [Zou & Hastie 05]) together with L-BFGS [Nocedal & Wright 99] and an efficient implementation of the forward-backward calculation summarizing transitions without active features. The authors of [Pal & Sutton$^+$ 06] have proposed to use beam-pruning [Ney & Mergel$^+$ 87] together with CRFs. This gives the opportunity to study the transition from unconstrained to an approximated training as we do in this publication. We will integrate elastic-net with RPROP, an easy to implement and memory efficient optimizer, and use sparse forward-backward together with a beam-pruning strategy using an efficient finite state transducer representation.

The calculation of the conditional log-likelihood (Equation (4.5)) in training of CRFs has a memory consumption proportional to the number of parameters λ_r, for which their derivative $\partial L/\partial\lambda_r$ is calculated, and the computational complexity can be characterized by a polynomial in $|\mathbb{Y}|$ with the degree of the largest target-n-gram δ. Computation time and memory consumption in search and training are similar. Unconstrained CRFs already have a huge demand in computational resources on comparatively small natural language processing tasks like grapheme-to-phoneme conversion. In the following section we will present an extension of a well known optimization algorithm to support L1 regularization to reduce the number of features where $\lambda_r \neq 0$ (Section A.1), followed by sections where we try to reduce the computation time. First, we will exploit that a large amount of target-n-grams $(y_i, y_{i-\delta}^{i-1})$ are not present in the training references and can safely be removed from the computation of the conditional log-likelihood (Equation (4.5)). Second, we reduce the amount of hypotheses y_1^I covered in Equation (4.5) by focusing on the hypotheses with largest feature description $H(y_{i-\delta}^n, A, x_1^J)$, known in the literature as a combination of dynamic programming and beam pruning ([Ney & Mergel$^+$ 87], Section 4.6.3). And finally, we will give a short description of the consequences of using joint-n-grams (Section 4.6.4).

4.6.1 RPROP-Elastic-Net-Extension

The parameters in the conditional random fields experiments are all optimized with the help of the resilient propagation (RPROP, [Riedmiller & Braun 93]). Each iteration m the parameters λ_1^R are updated based on the sign of the gradient times a step size $s_{r,m}$ per parameter. The size of

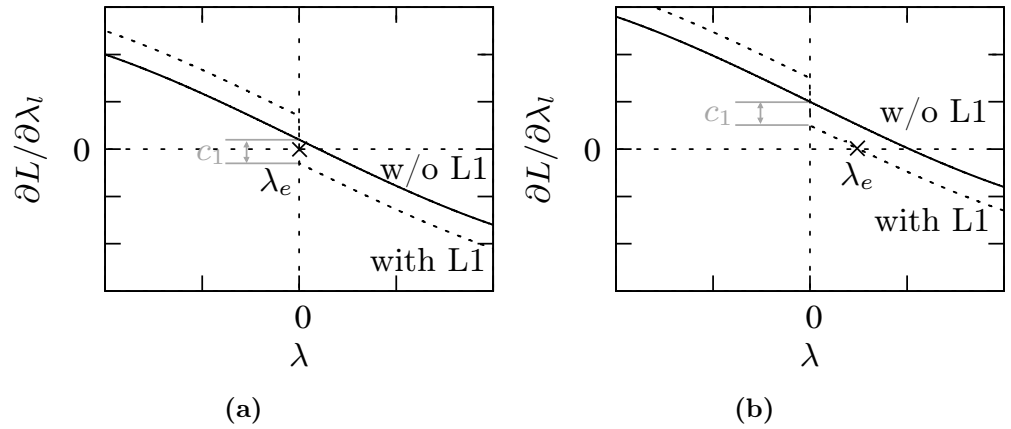


Figure 4.4 Sketches of the gradient of the objective function from Equation (A.1) with equal L1 regularization c_1 but different offset.

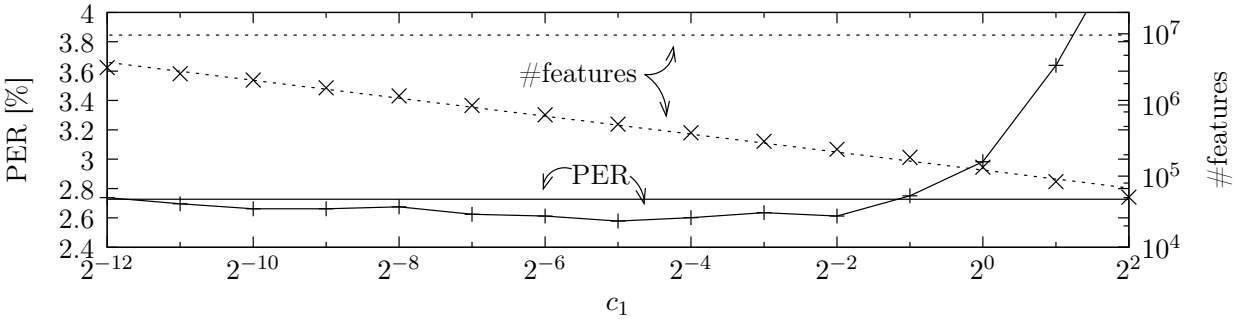


Figure 4.5 Effect of the L1 parameter c_1 on the performance on the CELEX dev set and the resulting number of features. The error rate plot is connected with a line to guide the eyes of the reader, while the number of features is connected by a log-log-linear regression $c_1^{-0.41} \cdot 122k$. The horizontal lines are the error rate and the respective number of features without L1 regularization. The number of features is reduced by a factor of 10 within 14 iterations. In the end at $c_1 = 1/4$ the error rate is 2.6% in phoneme error rate (PER) and $238k$ of the original $9.6M$ derived features are selected, while at $c_1 = 1/16$ the error rate is 2.6% in PER and $408k$ features are selected. (See Section 4.6.1)

the step size $s_{r,m}$ is defined by the last and next to last gradient. Including an L1-regularization $c_1||\lambda_1^R||_1$ in the conditional log-likelihood L (Equation (4.5))

$$\frac{\partial L'}{\partial \lambda_r} = \frac{\partial L}{\partial \lambda_r} - 2c_2\lambda_r - c_1 \operatorname{sign}(\lambda_r) \tag{4.10}$$

results in a jump in the gradient of L (curve "with L1" (dotted) in Figure 4.4). With $|\lambda_r| \gg 0$, the L1-regularization part of Equation (A.1) only changes the final optimum of λ_r by a small offset, but next to $\lambda_r = 0$ there are two distinct cases sketched in Figure 4.4:

- **a** The regular RPROP algorithm will set the final parameter $\lambda_e = 0$ after an infinite number of iterations.
- **b** It will trim it to the final parameter $\lambda_e \neq 0$.

In the appendixA.1, an extension of the RPROP algorithm, developed in the context of this thesis, recognises the case a already in one or two iterations without confusing it with case b.

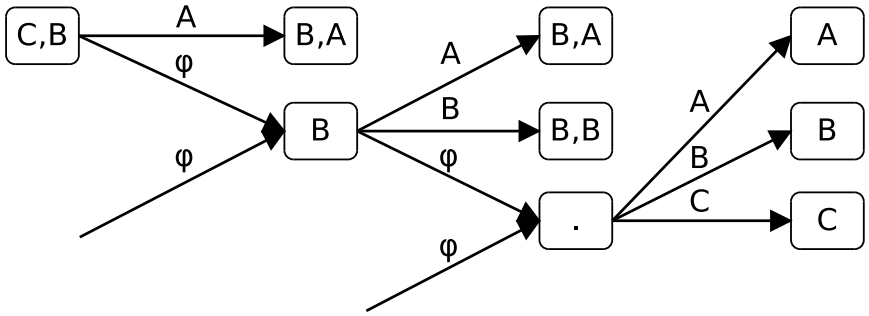


Figure 4.6 In the context of backing-off following the failure-transitions (φ) reduces the history saved with each state (most recent symbol on the right). In this example the history $(y_{i-2}, y_{i-1}) =$ (C,B) is reduced with the φ transition to the history (B) and the empty history. The maximum history length in this figure is $\delta = 2$. Every state can be reached from a different state with different histories or transitions. The algorithms in Figures 4.7, 4.8, and 4.9 are designed with this pattern in mind.

Experimental Results for the RPROP-Extension

To evaluate the effect of the L1 regularization parameter, the experimental setup 5 in Table 4.4 is taken and its L1-regularization parameter c_1 re-tuned, while the L2-regularization parameter c_2 is kept at $c_2 = 1/16 = 2^{-4}$. We only compute the derivatives of the features λ_r which have been seen in the references of the training corpus at least once (count cut-off of 1), selecting $9.6M$ of $71M$ features. The L1-parameter controls the number of $\lambda_r \neq 0$, affecting the performance of the system. Figure 4.5 demonstrates the resulting feature numbers and the performance with respect to PER. The error rate is mainly not affected in $0 \leq c_1 \leq 1/4$. Starting from this point, the performance drops strongly. A log-log-linear regression of $c_1^{-0.41} \cdot 122k$ of the final number of used features shows that the number of features can be smoothly selected. For further experiments, $c_1 = 1/16$ with a distance to the drop in performance is selected resulting in an error rate of 2.6% in PER and $408k$ features.

4.6.2 Scalable Training of N-grams

The extension of the RPROP algorithm to support L1-regularization reduces the amount of features selected during training. The following sections will focus on extensions reducing the amount of computation time. To keep things simple the respective equations will only cover conditional random fields without hidden variables (Equation (4.2)), but the final algorithm can be applied exactly in the same way to hidden conditional random fields.

As the sum in the denominator of Equation (4.2) covers all possible target sequences $\sum_{y_1^I}$, traditional linear chain conditional random fields systems include target n-gram features for every possible transition $y_{i-\delta}^{i-1} \to y_i$. In language modeling (LM), it is state-of-the-art to only model the transitions which are part of the reference of a given training corpus and else back-off to shorter transitions (cf. e.g. [Chen & Goodman 99]). In CRFs, the LM backing-off is useful in the calculation of the posterior scores in the update $D_{r,n}$ in Equation (4.5) of each feature. With a full matrix of target n-gram transitions, the posterior is as defined in Equations 3.10. Assuming for each possible history $y_{i-\delta}^{i-1}$ with length δ a set $S(y_{i-\delta+1}^{i})$ defining those y_i for which the feature functions h_r can be reduced to a shorter context $h_r(y_{i-\delta}^{i}, A, x_1^J) = h_r(y_{i-\delta+1}^{i}, A, x_1^J)$, a simplification described in [Lavergne & Cappé$^+$ 10] for target-bi-grams, can be applied. The sets $S(y_{i-\delta+1}^{i})$ are defined in our work with a feature count cut-off of 1. Only those features are used which are part of the training corpus. The simplification is called sparse-forward-backward. If only *target-bigrams* and

```
backward(G)
  Q ← dfs(G)
  for each q ∈ Q do
   β[q] ← β'[q] ← 0̄
  for each q ∈ Q do
   E ← ∅
   (β[q], β'[q]) ←
    bState(q, 0̄, E, 0)
  return β, β'
```

```
bState(q, r, E, l)
  for each e ∈ E[q] do
   if e ≠ φ then
    bArc(e, r, E)
    E ← E ∪ {e}
   else
    r, r' ←
     bFailArc(e, r, E, l)
   if q ∈ F then
    r ← r ⊕ 1̄
  return r, r'
```

```
bArc(e, r, E)
  if e ∉ E then
   r ← r ⊕ β[n[e]]
```

```
bArc'(e, r, E)
  if e ∈ E then
   r ← r ⊕ β[n[e]]
```

```
bFailArc(e, r, E, l)
  r' ← w[e]⊗
   bState(n[e], r, E, l + 1)
  r ← r ⊕ r'
  return r, r'
```

```
bFailArc'(e, r, E, l)
  if l = 0 then
   r' ← w[e] ⊗ (β[n[e]]⊖
    bState(n[e], r, E, l + 1))
   r ← r ⊕ r'
  else
   r' ← 0̄
  return r, r'
```

Figure 4.7 Single source shortest distance algorithm in backward orientation. The potentials β, β' are updated in the order as the depth first search finalizes the states of G, $dfs(G)$. For every state the method `bState` is called using the methods `bArc` and `bFailArc`, which calls `bState` recursively. The set $\mathcal{E}$ keeps track of already found transitions. The faster methods `bArc'` and `bFailArc'` can be used if the used semiring supports an operation $\ominus$ with $a \oplus b = c \Leftrightarrow a = b \ominus c$.

no alignment is used ($\delta = 1$) the forward potential (Equation (3.10)) is modified to:

$$\begin{aligned}\alpha_i(y_{i-1}) = &\sum_{y_{i-2} \in S(y_{i-1})} \exp\left(\sum_{r=1}^{R} \lambda_r h_r(y_{i-2}, y_{i-1}, x_1^J)\right) \alpha_{i-1}(y_{i-2}) \\ &+ \sum_{y_{i-2} \notin S(y_{i-1})} \exp\left(\sum_{r=1}^{R} \lambda_r h_r(y_{i-2}, y_{i-1}, x_1^J)\right) \alpha_{i-1}(y_{i-2}) \qquad (4.11) \\ = &\sum_{y_{i-2} \in S(y_{i-1})} \left(\exp\left(\sum_{r=1}^{R} \lambda_r h_r(y_{i-2}, y_{i-1}, x_1^J)\right) - \exp\left(\sum_{r=1}^{R} \lambda_r h_r(\cdot, y_{i-1}, x_1^J)\right)\right) \alpha_{i-1}(y_{i-2}) \\ &+ \exp\left(\sum_{r=1}^{R} \lambda_r h_r(\cdot, y_{i-1}, x_1^J)\right) \sum_{y_{i-2}} \alpha_{i-1}(y_{i-2}) \qquad (4.12)\end{aligned}$$

First, the sum is split into two partitions with the help of the set $S(y_{i-1})$, and second, the sum $\sum_{y_{i-2} \notin S(y_{i-1})}$ is extended to all indexes resulting in a correction of the other sum. Thus, the second

```
forward(G)
  Q ← dfs^{-1}(G)
  for each q ∈ Q do
   α[q] ← α'[q] ← 0̄
  α[0] = 1̄
  for each q ∈ Q do
   E ← ∅
   fState(q, α[q], E, 0)
  return α, α'
```

```
fState(q, r, E, l)
  for each e ∈ E[q] do
   if e ≠ φ then
    fArc(e, r, E, l)
    E ← E ∪ {e}
   else
    fFailArc(e, r, E, l)
```

```
fArc(e, r, E, l)
  if e ∉ E then
   r' ← w[e] ⊗ r
   α[n[e]] ← α[n[e]] ⊕ r'
  if l > 0 then
   α'[q][e] ← r
```

```
fArc'(e, r, E, l)
  if l = 0 then
   r' ← w[e] ⊗ r
   α[n[e]] ← α[n[e]] ⊕ r'
  else if e ∈ E then
   α'[q][e] ← r
```

```
fFailArc(e, r, E, l)
  r' ← w[e] ⊗ r
  fState(n[e], r', E, l + 1))
```

```
fFailArc'(e, r, E, l)
  if l = 0 then
   r' ← w[e] ⊗ r
   α[n[e]] ← α[n[e]] ⊕ r'
   fState(n[e], r', E, l + 1))
```

Figure 4.8 Single source shortest distance algorithm in forward orientation. The potentials α, and transition wise potentials α' are updated in the inverted order a depth first search finalizes the states of G, $dfs(G)^{-1}$. For every state the method `fState` is called using the methods `fArc` and `fFailArc`, which calls `fState` recursively. The set $\mathcal{E}$ keeps track of already found transitions. The faster methods `fArc'` and `fFailArc'` can be used if the used semiring supports an operation $\ominus$ with $a \oplus b = c \Leftrightarrow a = b \ominus c$.

sum can be precomputed and only the calculation of the first sum with respect to the small set $S(y_{i-1})$ is needed for each history y_{i-2}, y_{i-1}.

We will extend this bigram computation time reduction to arbitrary target n-grams. To model the backing-off for LMs in finite state transducers (FSTs) [Allauzen & Riley$^+$ 07], an extension to finite state transducers is described in [Allauzen & Mohri$^+$ 03]: the φ- or failure-transition. It is a variant of the ϵ-label excluding successing transitions, already used in the state emitting the failure transition. E.g. in Figure 4.6 after visiting the state with history (y_{i-2}, y_{i-1}) = (C,B) and following the φ transition by definition a state is reached with reduced history. In this state, the transition `A` can only be used when coming from a state with different outgoing arcs. With respect to LMs, the state with history `(C,B)` emits the trigram `(A|C,B)`. To generate the bigram `(B|B)`, the φ-transition is applied (representing a reduction to the history `(B)`) and the transition `B` is used. The described restriction after φ-transitions is covered in OpenFST [Allauzen & Riley$^+$ 07] with use of filters in composition and single best search.

Inspired by backig-off in LMs, Equation (4.12) has been implemented with a state for every possible history $y_{i-\delta}^{i-1}$ and states for the shorter histories $y_{i-\delta-1}^{i-1}, \ldots$ until the empty history. A regular transition is used if $y_i \in S(y_{i-\delta}^{i-1})$ and a φ-transition to the shorter history otherwise. The

```
posterior(G)
  α, α' = forward(G)
  β, β' = backward(G)
  for each q ∈ Q do
   for each e ∈ E[q] do
    if e ≠ φ then
     β'' ← β[q]
    else
     β'' ← β'[q]
    default: α'' ← α[q] ⊕ α'[q][e]
    if *arc': α'' ← α[q] ⊖ α'[q][e]
    w[e] ← α'' ⊗ w[e] ⊗ β'' ⊗ (β[0])^{-1}
```

Figure 4.9 Final posterior algorithm using `forward` and `backward`. The command after "if `arc'`" may be used instead of the command after "default" if `xArc'` and `xFailArc'` have been used.

first sum in Equation (4.12) is estimated at the full history state and the precomputed sum is estimated at the shorter history state. Using filters to dynamically remove the φ-transitions would not reduce the computation time as states with shorter histories would be removed, too, and the precomputation of the second sum in Equation (4.12) could not used anymore. Thus, we propose the algorithms presented in Figures 4.7, 4.8, and 4.9. They utilize computation time improvements deduced from a semiring supporting an operation $\ominus$ with $a \oplus b = c \Leftrightarrow a = c \ominus b$ (e.g. possible with the log-semiring, see e.g. Table 1 in [Mohri 09] for details of semirings). `Forward` and `Backward` calculations are constructed independently, but similarly. Restrictions with respect to the failure-/φ-transitions are tracked in the set $\mathcal{E}$, including all already seen input/output-label combinations. E.g. in the case of `backward(G)`, for every state the method `bState()` is called which recursively calls `bState()` for failure-/φ-transitions via the `bFailArc()` or `bFailArc'()` method. The `X'` methods are used, if the applied semiring supports an operation $\ominus$ with $a \oplus b = c \Leftrightarrow a = c \ominus b$, reducing the computational cost, since the recursion of `state` is restricted to one level. In addition to the standard potentials, back-off weights β' per state and additional forward weights α' per transition have to be kept in memory. The presented algorithm does not support cycles, which are not needed for the presented modeling of (hidden) conditional random fields.

Experimental Results for N-grams

Tables 4.6 and 4.7 include the final results combining all presented conditional random fields extensions. Especially they include the results using `target-2-grams` and `target-3-grams` using the algorithms presented in the last section. The longer target-n-grams improve mainly the WER (WER) by 6% rel. on CELEX and 4% on PRONLEX, with an increase of a factor 15 and 17 respectively in computation time. We count the computation time spent in training by summing up the computation time spend by all parallel processes in all iterations. A training without the speed up described in this section lasted $133h$ with target-2-grams and $6770h$ with target-3-grams. The computation time in our implementation is dominated by the number of histories $y_{i-\delta}^{i-1}$, which are mainly not changed with target-2-grams, as $S(y_{i-1}) \approx \mathbb{Y}$. However, with target-3-grams the history is a tuple and $|S(y_{i-2}, y_{i-1})| \ll |\mathbb{Y} \times \mathbb{Y}|$.

4.6.3 Pruning in Training

The decision for the best hypothesis (Equation (4.9)) can be decomposed for every position j into:

$$\underset{\tilde{t}^j_{j-\delta}}{\operatorname{argmax}} \left\{ \alpha_j^{max}(\tilde{t}^j_{j-\delta}|x_1^J) + \beta_{j+1}^{max}(\tilde{t}^j_{j-\delta+1}|x_1^J) \right\} \tag{4.13}$$

$$\alpha_j^{max}(\tilde{t}^j_{j-\delta}|x_1^J) = \underset{\tilde{t}_1^{j-\delta}}{\operatorname{argmax}} \left\{ \sum_{\tilde{j}=1}^{j} \sum_{r=1}^{R} \lambda_r h_r(\tilde{t}^{\tilde{j}}_{\tilde{j}-\delta}, x_1^J) \right\}$$

$$\beta_{j+1}^{max}(\tilde{t}^j_{j-\delta+1}|x_1^J) = \underset{\tilde{t}^I_{j+1}}{\operatorname{argmax}} \left\{ \sum_{\tilde{j}=j+1}^{I} \sum_{r=1}^{R} \lambda_r h_r(\tilde{t}^{\tilde{j}}_{\tilde{j}-\delta}, x_1^J) \right\}$$

Combining this dynamic programming equation with beam pruning reduces the search space with a pruning threshold τ to

$$\underset{\tilde{t}^j_{j-\delta} \in \Gamma_j}{\operatorname{argmax}} \left\{ \alpha_j^{max}(\tilde{t}^j_{j-\delta}|x_1^J) + \beta_{j+1}^{max}(\tilde{t}^j_{j-\delta+1}|x_1^J) \right\} \tag{4.14}$$

$$\Gamma_j = \left\{ \tilde{t}^j_{j-\delta} : \alpha_j^{max}(\tilde{t}^j_{j-\delta}|x_1^J) \geq \max_{\tilde{t}^j_{j-\delta}} \left\{ \alpha_j^{max}(\tilde{t}^j_{j-\delta}|x_1^J) \right\} - \tau \right\},$$

which is a well known pruning strategy already published for automatic speech recognition in [Ney & Mergel$^+$ 87]. Please note that the computation time speed up is achieved by omitting the backward potential $\beta_{j+1}^{max}(\tilde{t}^j_{j-\delta+1}|x_1^J)$. In a hidden conditional random fields setup (e.g. Table 4.4), there exists an alternative path utilizing skip penalties. At a state emitting begin, inside, and skip labels (bottom of Figure 4.3), there are three arcs for every target symbol, with weights $\delta_0 + H(t^{\tilde{j}}_{\tilde{j}-\delta}, A, x_1^J)$ (inside), $\delta_1 + H(t^{\tilde{j}}_{\tilde{j}-\delta}, A, x_1^J)$ (begin), $\delta_2 + H(t^{\tilde{j}}_{\tilde{j}-\delta}, A, x_1^J)$ (skip). Considering only these weights without the following weights in the skip state, the training can remove the wrong hypotheses, as the often negative δ_2 will be compensated by an often positive δ_0 in the skip state. To take future scores into account, the `Push`-Algorithm [Mohri 09] can move the maximum future score as far as possible to the initial state. Thus, Γ_j in Equation (4.14) for states emitting begin, inside, and skip labels is changed to

$$\Gamma_j = \left\{ \tilde{t}^n_{j-\delta} : \alpha_j^{max}(\tilde{t}^j_{j-\delta}|x_1^J) + \hat{\beta}_{j+1}^{max}(\tilde{t}_j|x_1^J) \right.$$

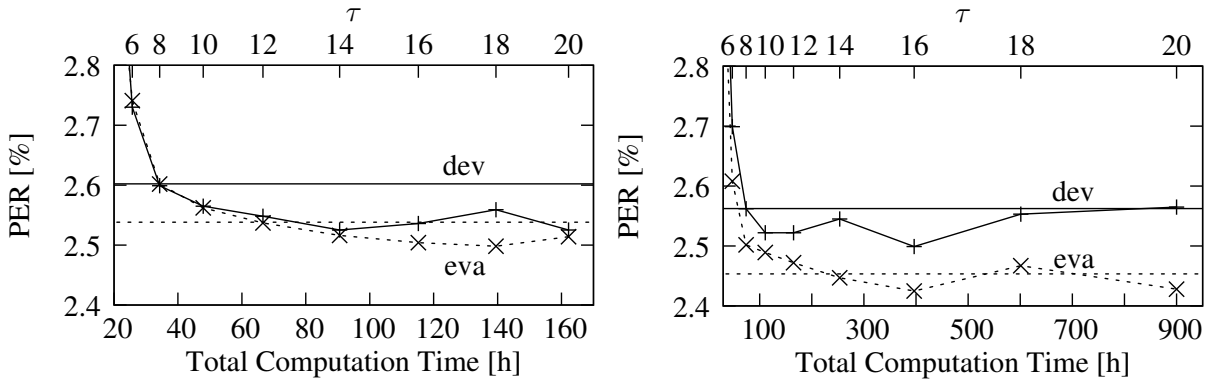
$$\left. \geq \max_{\tilde{t}^j_{j-\delta}} \left\{ \alpha_j^{max}(\tilde{t}^j_{j-\delta}|x_1^J) + \hat{\beta}_{j+1}^{max}(\tilde{t}_j|x_1^J) \right\} - \tau \right\}, \tag{4.15}$$

with $\hat{\beta}_{j+1}^{max}(\tilde{t}_j|x_1^J)$ taking all features into account except target-n-grams and joint-n-grams.

To also reduce the computation time of the parameter estimation, we propose to use Γ_j additionally in a modification of the conditional log-likelihood (Equation (4.5)). The summation in $D_{r,n}$ (Equation (4.7)) is constrained using Γ_j

$$D^*_{r,n} = \sum_{j=1}^{J} \sum_{\tilde{t}^j_{j-\delta} \in \Gamma_j} p(\tilde{t}^j_{j-\delta}|x^J_{1,n}) h_r(\tilde{t}^j_{j-\delta}, x^J_{1,n}). \tag{4.16}$$

In Section 3.2.4, it is described that the numerator statistics $N_{r,n}$ are extracted from the final transducer by composing a reference acceptor. Pruning can potentially remove the numerator. This is detected and the pruning threshold τ is increased for the respective utterance.



(a) target-2-grams **(b)** target-3-grams

Figure 4.10 Results of beam-pruning applied in model estimation. (a) use bigrams as target-n-gram features, while (b) use bigrams and trigrams as target-n-grams features. Solid lines connect results on the development set, and dashed lines connect the results on the evaluation set. The horizontal lines represent the result without beam-pruning. Computation time without beam-pruning is $137h$ for (a) and $2364h$ for (b). The top boundaries of the plots report the used pruning threshold τ. (See Section 4.6.3)

Experimental Results for Pruning in Training

The experiments presented in Figure 4.10 show the transition from strong approximations to the model estimation to a full hypotheses space. Starting from $\tau = 10$ the training is stable against the pruning and the computation time is largely reduced without a degradation in recognition performance. E.g., with a pruning threshold of $\tau = 14$ we have a save and stable calculation time improvement down to 2/3 of the original training time for target-bigrams and to 1/10 of the original training time for target-bigrams together with target-trigrams. However, for low τ, the training gets instable with a large increase in PER. Some approaches for large tasks, like for statistical machine translation [Blunsom & Cohn$^+$ 08], tried to approximate conditional random fields by similarly pruned search spaces. They needed to havily prune the system to keep the computation times in resonable regions. Maybe they suffered from the same instabilities we see here for small τ.

In the first iterations, the beam-pruning has only a small effect on the computation time as all possible paths are equally likely. Thus, we let the estimation of target-n-gram features start after the source-n-gram features have already been updated for two iterations without pruning.

4.6.4 Joint-N-Grams

In [Jiampojamarn & Cherry$^+$ 10], the authors propose to use joint-n-gram features. Joint-n-gram features are a composition of a source- and a target-n-gram $(y_{i-\delta}^{i}, x_{A(i)-\gamma_1}^{A(i)+\gamma_2})$. Actually, these features represent phrases as in phrase-based translation [Koehn 10], but are allowed to overlap. In our implementation, the joint-n-gram features are applied after applying all other features and steps described in Figure 3.2. The parameters λ_r of the joint-n-gram features are added to the weights of the final FST by evaluating which source-n-grams and which target-n-grams already have been applied. This reduces feature look-ups drastically, while additionally accepting the restriction that only joint-n-grams may be applied where the respective source-n-grams and target-n-grams are used. Using all possible combinations $(y_{i-\delta}^{i}, x_{A(i)-\gamma_1}^{A(i)+\gamma_2})$ with the configuration in Section 4.3 $(\gamma_1, \gamma_2 = 0, \ldots, 5)$ would result in a huge amount of features. Elastic-net can be used to shrink this set of features. We trained all features besides the joint-n-grams for 20 iterations using

elastic-net. Starting from iteration 21, only those joint-n-gram features are estimated where the λ_r of the source-n-gram and the $\lambda_{r'}$ of the corresponding target-n-gram are both not equal to zero ($\lambda_r \neq 0 \neq \lambda_{r'}$) in iteration 20. This strategy is sufficient to combine target-2-grams with all source-n-grams. However, with using longer target-n-grams, memory is exhausted during the feature selection process. To overcome this issue, we tried to limit the joint-n-grams to target-2-grams and did not combine the used longer target-n-grams in the joint-n-grams. In a second approach, we evaluated to limit the total size of the joint-n-gram to six symbols on source and target side together.

Experimental Results for Joint-N-Grams

Table 4.5 presents the results extending a baseline (line 1) selected from the final feature build-up in Table 4.6 with joint-n-gram features. First, a target-2-gram baseline is extended with all possible joint-n-grams, where $\lambda_l \neq 0 \neq \lambda_{l'}$ in iteration 20, resulting in an improvement from 12.4 to 12.0 in WER on the development set. Lines 4,5 and lines 8,9 answer the question if this improvement can be generalized to longer target-n-grams. Here, a different behavior on the development and the evaluation set can be identified. While on the development set joint-n-grams help in both cases, and target-4/5-grams did not help, the situation on the evaluation set is opposite. Additionally using target-3/4/5-grams in joint-n-grams did not yield an improvement.

Table 4.5 Joint-n-gram build-up on CELEX (cf. Section 4.6.4)

	features	# features final/max	DEV PER [%]	DEV WER [%]	EVAL PER [%]	EVAL WER [%]
1	Raw 8 of Table 4.6	400k/9.61M	2.5	12.4	2.5	12.2
2	+ joint-n-grams	759k/13.5M	2.5	12.0	2.5	11.8
3	+ target-3-grams	393k/9.62M	2.5	12.2	2.4	11.7
4	+ joint-n-gr. (2-gr)	642k/11.3M	2.5	**11.6**	2.3	**11.0**
5	+ joint-n-gr. (<6)	590k/12.2M	2.5	11.7	2.4	11.2
6	+ target-4-grams	399k/9.67M	2.6	12.2	2.4	11.5
7	+ target-5-grams	415k/9.75M	2.6	12.1	2.4	11.3
8	+ joint-n-gr. (2-gr)	696k/9.19M	2.6	11.8	2.4	11.2
9	+ joint-n-gr. (<6)	749k/13.5M	2.6	11.7	2.4	11.2

4.7 Final System

All reported approaches are compared in Table 4.6 for the CELEX corpus and in Table 4.7 for the PRONLEX corpus. From Table 4.6 and Table 4.7 it can be concluded that the most important features are long context source-n-grams, followed by target-2-grams. Longer contexts over target symbols (target-n-grams) and combinations of the source and target context bridge the gap to the baselines, finally approaching the work of [Jiampojamarn & Cherry$^+$ 10] on CELEX. Significance figures are calculated according to the procedure in [Parihar & Picone 02] (Appendix B). All three systems (lines 1, 2, and 11 in Table 4.6) are not significantly different in WER wrt. a significance level of 5%. However, on PRONLEX our approach is significantly better in WER with a level of 2% (line 10 compared to line 1).

Error analysis have shown that for some of the remaining errors even longer contexts may be useful. E.g. the *e* in the word *d<u>e</u>mographic* has been hypothesized like in *d<u>e</u>mocracy* but should be spelled as *p<u>e</u>ppermint* according to the reference. Here, the source-4-gram of the first four letters are equal, and thus confusing the model.

Table 4.6 Feature build-up on CELEX (cf. Section 4.7). The linear chain conditional random fields (LCCRFs) make use of an external segmentation extracted from recognition results gained with [Bisani & Ney 08].

	system/features	Training time [h]	DEV PER[%]	DEV WER[%]	EVAL PER[%]	EVAL WER[%]
1	[Jiampojamarn & Cherry$^+$ 10]					10.8
2	joint n-grams [Bisani & Ney 08]				2.5	11.4
3	LCCRFs (same feat. as 7)	38	2.7	13.1	2.6	12.9
4	LCCRFs (same feat. as 9)	744	2.7	12.7	2.5	12.1
5	HCRFs single + (δ_j)	4	52.5	97.1	52.7	97.7
6	HCRFs + source n-grams	15	4.0	20.9	3.8	20.2
7	HCRFs + target-2-grams	137	2.6	12.6	2.5	12.3
8	HCRFs + prune ($\tau = 14$)	90	2.5	12.4	2.5	12.2
9	HCRFs + target-3-grams	2364	2.6	12.3	2.5	11.6
10	HCRFs + prune ($\tau = 14$)	253	2.5	12.2	2.4	11.7
11	HCRFs + joint-n-grams	1169	**2.5**	**11.6**	**2.3**	**11.0**

Table 4.7 Feature build-up on PRONLEX (cf. Section 4.7)

	system/features	Training time [h]	DEV PER[%]	DEV WER[%]	EVAL PER[%]	EVAL WER[%]
1	joint n-grams [Bisani & Ney 08]				6.8	27.3
2	single + (δ_j)	15	49.1	95.7	49.4	96.2
3	+ source n-grams	49	23.5	82.3	23.5	82.5
4	+ target-2-grams	417	6.3	26.9	6.4	27.1
5	+ prune ($\tau = 14$)	249	6.3	27.1	6.4	26.7
6	+ joint-n-grams	814	6.3	26.8	6.3	25.8
7	+ target-3-grams	756	6.3	6.3	26.5	26.1
8	+ target-4-grams	1547	6.5	26.8	6.4	25.6
9	+ target-5-grams	3101	6.4	26.1	6.4	25.4
10	+ joint-n-gr.	10976	**6.3**	**25.9**	**6.3**	**25.5**

4.8 Analysis of the Search Space

The research in this section was conducted in the framework of the Quaero program as a joint work with LIMSI. Objective of this research was to contrast the approach presented in the remainder of this chapter against the approach in [Lavergne & Allauzen$^+$ 11], applied there on statistical machine translation. Other than our hidden conditional random fields, the approach in [Lavergne & Allauzen$^+$ 11] scales to the large vocabularies, feature sets, and training corpora used in machine translation. The main difference in the approaches, found in a detailed analysis of the software, was in the structuring of the hidden latent variable applied in the hidden conditional random fields. Hidden conditional random fields (Equation 4.3) can be reformulated as

$$p(y_1^I|x_1^J) = \sum_{A\in\mathbb{A}} p(y_1^J, A|x_1^J) = \frac{\sum\limits_{A\in\mathbb{A}} \exp\left(\sum\limits_{n=1}^{N}\sum\limits_{r=1}^{R} \lambda_r h_r(A, y_{n-\delta}^n, x_1^M)\right)}{\sum\limits_{\tilde{A}\in\mathbb{A}}\sum\limits_{\tilde{y_1^J}} \exp\left(\sum\limits_{n=1}^{N}\sum\limits_{r=1}^{R} \lambda_r h_r(\tilde{A}, \tilde{y}_{n-\delta}^n, x_1^M)\right)} \tag{4.17}$$

Here, $A \in \mathbb{A}$ emphasizes that the segmentation A is constrained by some set $\mathbb{A}$, which is implicitly or explicitly defined by the used hidden conditional random fields.

4.8.1 Three Ways to Cope with Hidden Structure

Estimating parameters with respect to Equation 4.17 without constraining set $\mathbb{A}$ is infeasible. However, examples from grapheme-to-phoneme conversion suggest that the segmentations may be restricted to some extent. E.g. [Bisani & Ney 08] used in their final system only graphones with size zero or one for source- and target-n-grams. In Section 4.4, our hidden conditional random fields are already contrasted with linear chain conditional random fields (LCCRF) estimated and evaluated on utterances preprocessed with the software of [Bisani & Ney 08]. Thus, the problem is recast as a sequence labeling task with 1-to-1 alignments.

Our hidden conditional random fields implementation is summarized in Figure 4.11 (as a reduction of Figure 3.2 and Figure 4.3) and uses phonemes annotated with special segmentation labels inspired by the *BIO* scheme [Ramshaw & Marcus 95] organized as in hidden Markov models (HMMs), leaving the modeling of the phrase information to phrase-like features (Section 4.3 for more details). First, the grapheme sequence is represented as a chain of symbols in a finite state transducer ([Mohri 09], Figure 4.11a). Second, each arc is duplicated for each possible phoneme symbol and weighted with the features taking only one phoneme symbol into account (Figure 4.11b). In a third step, all the phoneme symbols are extended with three labels indicating the beginning of a new phoneme _ *b*, its continuation _ *c*, and the doubling of a phoneme that implies the beginning of a phoneme _ *d*. The arcs labeled with the doubling label _ *d* start an alternative path with two arcs per phoneme (Figure 4.11c). The final result is composed with an n-gram acceptor weighted with the features taking only the target-n-grams on the phonemes into account, and in a last step all remaining features are applied. As within hidden Markov models, the arcs indication continuation/beginning/doubling are weighted with a penalty $\delta_0/\delta_1/\delta_2$ (designed as hidden conditional random fields features). As a consequence of the begin, continuation, and doubling labels each symbol in the source sequence is aligned to one or two symbols in the target sequence, while at the same time the opposite alignment of aligned source symbols with respect to the target symbols is known, too. Thus, our implementation may be seen as a word segmentation as sketched in Figure 4.12a.

The third approach is a hidden conditional random fields implementation inspired by phrase based decoding. Critical part is a conversion table from source-n-grams to target-n-grams, called phrase table in statistical machine translation. While in the word based approach, before the set constraining

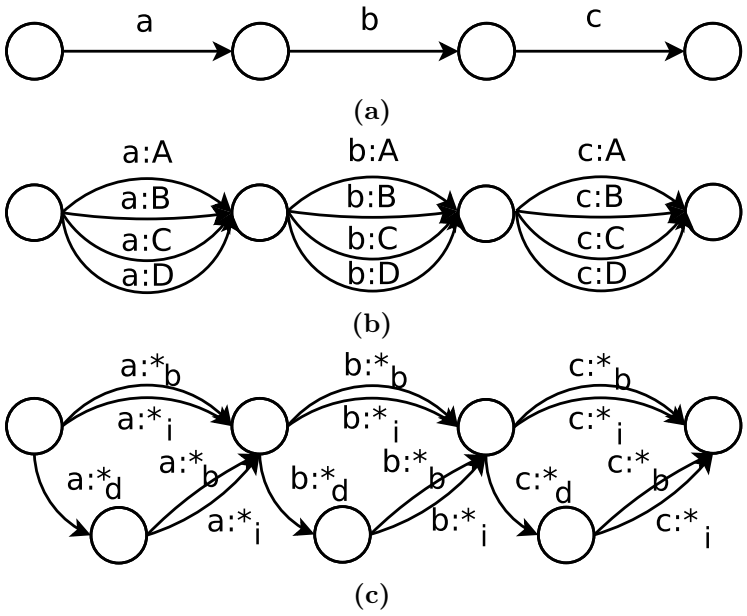
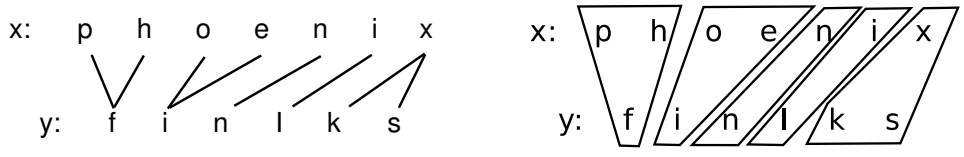


Figure 4.11 In the approach presented in the predecessor sections the source symbol sequence is represented as a chain (a), the input chain is augmented by the target vocabulary (`A`, `B`, `C`, `D`) and weighted by prior and source-to-target features (b). To support segmentations s every target symbol is extended with a label representing the beginning _ *b*, the continuation _ *c*, and a doubling of a source label and beginning of a new target label _ *d*. In case the doubling was selected, an alternative path is used with two arcs per source symbol (c).



(a) word segmentation **(b)** phrases

Figure 4.12 The segmentation of the grapheme-phoneme pair of the word "phoenix" as a word segmentation and b phrase segmentation.

the segmentations $\mathbb{A}$ was implicitly defined by the rules on top of the begin/continuation/doubling labels, the conversion table explicitly defines all possible segmentations of a source-target-sequence pair. This conversion table is constructed by the usage of an external method defining alignment points between the source and target sequence. In the experiments reported, GIZA++ [Och & Ney 03] was used, and the conversion table was extracted with the *grow-diag-final-and* heuristic [Koehn & Och$^+$ 03] with the maximum size of segments limited to 3. The conversion table is used in a pipeline of finite state transducer operations (Figure 4.13). Again, the input is represented as a chain of the input symbols. This input acceptor is composed with a segmentation transducer determining all possible segmentations of the source sequence, followed by a composition with the conversion table. The final transducer may be used with hidden conditional random fields features in inference or in model estimation of these features. All transducer operations were executed with the OpenFST library [Allauzen & Riley$^+$ 07], except for the forward-backward algorithm and the hidden conditional random fields integration relying on an in-house implementation. Features are constructed similarly to the word based implementation directly on the surface form as letters and phonemes. However, different to the word based segmentation, the smallest units are not single letters and phonemes but n-grams of letters and phonemes defined by the conversion table. Features similar to the source-n-gram features in Section 4.3 are combinations of the current phrase together with the source part of the predecessor and/or the successor phrase. The average size of a source n-gram is 1.84, letting this features span over $3 \times 1.84 = 5.52$ source symbols similar to the word based implementation. These source-n-gram phrase features are combined with features similar to the target-n-grams in Section 4.3 taking the target part of a phrase in context to the target part of the predecessor phrase. Both features are combined via and-combinations to joint-n-gram phrase features. To support longer contexts it is possible to compose the finally scored result with a language model acceptor. Through the constraint of the conversion table, some references are not reachable in model estimation. This is solved by replacing in these cases the reference by an oracle hypothesis selected with respect to the BLEU score [Papineni & Roukos$^+$ 02].

In both approaches, a sum over the segmentations is utilized in model estimation. In inference, this sum is commonly replaced by a maximization resulting in an asymmetry between training and testing. As two different segmentations in the phrase based approach are considered with different features, it can be expected that a sum in inference would be beneficial especially for this approach. However, having a sum within the maximization of the inference is infeasible as it involves the determination of a large automaton. So a 400-best list is extracted and determination is done by summing up the probabilities of all entries generating the same hypotheses.

4.8.2 Experimental Results

All three segmentations are contrasted together with the two baselines [Bisani & Ney 08] and [Jiampojamarn & Cherry$^+$ 10] with respect to conversion error rates in Table 4.8. Linear chain results are generated with the word based hidden conditional random fields software excluding the hidden extensions of Section 4.4. During model estimation the source and reference are 1-to-1 aligned with the help of the BIO-scheme and the external method from [Bisani & Ney 08], while in inference, the method in [Bisani & Ney 08] is only used to double each source symbol. The results for linear chain conditional random fields are presented with target-2-grams in line 3 and with target-3-grams in line 4 of Table 4.8. Including hidden structure and begin/continue/doubling features in exactly the same system gives in average an absolute improvement of 0.1% in PER and 0.5% in WER. As the search in the system is maximizing the phoneme sequence and the alignment together, we have distinct n-best entries for the same phoneme sequence with different alignments (consequence of the approximation for Equation 4.6). We extracted a 400-best list and combined the probabilities of all entries with equal phoneme sequence and reference the result as *determination* in Table 4.8 (line 9). However, including determination does not improve conversion

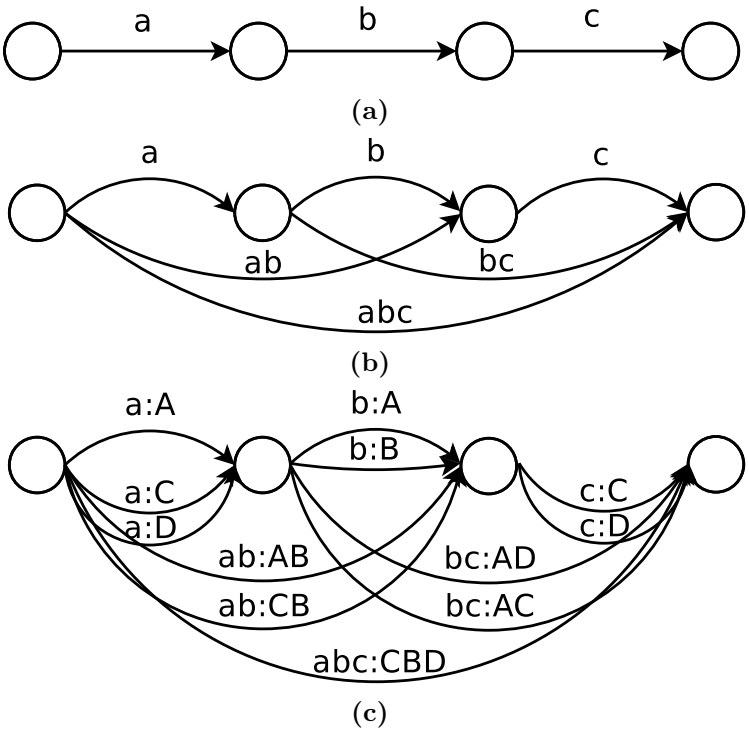


Figure 4.13 Phrase segmented system. As in the word based system in the remainder of this chapter the utterance is represented in the phrase based system with an acceptor a, which is composed with a segmentation transducer to consider all segmentations supported in the conversation table b. The result is composed with the conversion table to include all conversions c, and the final result is weighted with the HCRF features.

Table 4.8 Results on the CELEX corpus. Lines 1 and 2 provide baseline results where [Bisani & Ney 08] is the best found generative approach, and [Jiampojamarn & Cherry$^+$ 10] the best found discriminative approach on this task. Lines 3 and 4 provides a result for a system leaving the modulation of the segmentation to a the model [Bisani & Ney 08] and using the same features as in 7 and 8 except segmentation specific features. The next two blocks describe the build-up of the system A and system B including their best result (Section 4.8.2).

		EVAL	
		PER[%]	WER[%]
1	[Jiampojamarn & Cherry$^+$ 10]		10.8
2	joint n-grams [Bisani & Ney 08]	2.5	11.4
	LCCRF with alignments from [Bisani & Ney 08]		
3	- target-2-grams (same features as 7)	2.6	12.9
4	- target 3-grams (same features as 8)	2.5	12.1
5	$(e_{a_j}, f_j) + (\delta_j)$	52.7	97.7
6	+ source n-grams	3.8	20.2
7	word segmented + target-2-grams	2.5	12.3
8	+ target-3-grams	2.5	11.6
9	+ determination	2.5	11.7
10		3.2	14.5
11	phrase segmented + determination	3.1	14.1
12	+ 4-gram LM	3.1	14.0

errors, which is expected as paths with different segmentation but equal hypothesis have almost the same active features. Additionally, most of the time $p(y_1^J, A|x_1^J) \leq 50\%$ for the first best in the 400-best list. Only a small amount of the probability mass could change the conversion result.

Starting from line 10 in Table 4.8, the phrase based conversion results are presented. The results include more errors, and even as determination and composition with a 4-gram language model acceptor improve results, the accuracy of the word based system could not been reached. However, model estimation time and conversion time are drastically faster. Due to different architectures an exact number is hard to determine. The factor is expected in the range of 100-1000.

The difference in computation time is expected to be influenced by the structure of the respective search space. For the word *aback*, the word based system consider a search space of 13274 nodes and 207,356 arcs that corresponds to 1.74×10^{17} paths. For the same word, the phrase based system only explores 36,686 paths with 93 nodes and 1042 arcs. For the word *bent*, the word based system consider a search space of 10,336 nodes and 155,936 arcs that corresponds to 6.22×10^{13} paths, while the corresponding search space explored by the phrase based system contains 7914 paths made of 92 nodes and 1016 arcs. The huge reduction of the search space explains why the phrase based system is more than a hundred times faster than the word based system for model estimation and inference. While the phrase based system only permits a restricted set of conversions, the word based system includes all possible target sequences, which are not more than a factor of 2 larger than the source sequence. Moreover, while the word based system tends to select for the test set 1-to-1 alignments (in 80% of the cases) and 2-to-1 alignments (in 17% of the cases), the repartition for the phrase based system differs: 36% of 1-to-1, 29% of 2-to-2, 17% of 2-to-1, 10% of 3-to-2, 5% of 3-to-3 and 1% of 2-to-3 (the others can be neglected).

To assess whether this restriction of the search space may explain the decrease in performance, the oracle hypothesis is estimated for the test set with respect to the Bleu score [Papineni & Roukos$^+$ 02]. The oracle hypothesis exhibits a Per of 0.3% and a Wer of 1.3%. These oracle error rates suggest that the restricted search space is not the cause of the degradation in performance. As the search space seems to be sufficient and the same training algorithm was applied, we expect that the differences in the feature set resulted in the difference in the performance. Within the phrase based approach the features were constrained to the phrase boundaries. It lacked features for phonemes within the phrase boundaries conditioned on characters from the predecessor and successor phrases.

4.9 Conclusion

In this chapter, we reported our work on conditional random fields for grapheme-to-phoneme conversion. We first showed that an integrated training of the segmentation as a hidden variable to conditional random fields elevates performance. Integrating longer target contexts and combinations of source and target context give rise to efficient strategies, including sparse-forward-backward, elastic-net, and pruning in training. All of them have been carefully designed and the final algorithms are included in this chapter. Through the use of these strategies, computation time and memory consumption of complex feature combinations could be drastically reduced. The final result is competitive as it is placed between the two best known baselines on this task.

Additionally, we have compared this approach to a baseline designed for statistical machine translation to estimate the potential and the strength of both approaches. Final result was that the approach presented in the first part of this chapter has a high accuracy but is restricted by its high demand of computational power.

The scientific contributions of this chapter have already been published in the conference proceedings [Lehnen & Allauzen$^+$ 13, Lehnen & Hahn$^+$ 12, Lehnen & Hahn$^+$ 11b, Heigold & Hahn$^+$ 11, Lehnen & Hahn$^+$ 11a, Hahn & Lehnen$^+$ 11].

The work presented in this chapter was the work of multiple authors. The contributions of the different authors can be summarized as:

- An efficient HCRF implementation adopting the BIO scheme.
  Idea by *Lehnen* and first implementation was done by *Guta*. This implementation was verified by a re-implementation done by *Lehnen* and *Hahn*
- Elastic-net implemented within RPROP.
  The algorithm was designed by *Lehnen*. The software implementation and experiments were done together by *Lehnen* and *Hahn*.
- Sparse-forward-backward for arbitrary n-gram sizes.
  The idea and equations on bigram level were first done by *Ney* and *Lavergne*[1]. The algorithm and the implementation for arbitrary n-grams was done by *Lehnen*.
- Beam pruning in HCRF training.
  All implementation and experiments were done by *Lehnen*.
- Analysis of the influence of joint-n-grams.
  This was joint work between *Lehnen* and *Hahn*.
- Analysis of the search space.
  The original idea for this work came from *Yvon*[2] and *Ney*. The experiments were done, analysed and presented by *Lehnen*, *Allauzen*[3], and *Lavergne*[1].

If not stated otherwise, the authors were part of RWTH Aachen University during the mentioned research.

[1] Thomas Lavergne, University Paris-Sud and LIMSI/CNRS - Information, Written and Signed Language group
[2] François Yvon, University Paris Sud, Director of the LIMSI, Researcher at LIMSI/CNRS
[3] Alexandre Allauzen, University Paris-Sud and LIMSI/CNRS - Spoken Language Processing group

5. Applying Maximum Entropy Approaches to Machine Translation

In this chapter we will report two different approaches of using maximum entropy training in statistical machine translation. The first approach will adapt the model of [Goodman 01] only used with neural network training before [Le & Allauzen$^+$ 12] to conditional random fields. The motivation is to solve the computation time issues of conditional random fields with statistical machine translation, while keeping the accuracy of the conditional random fields (Section 5.1). As conditional random fields are based on a {0,1} loss on sentence level, they are very sensitive to the reference translation. However, in statistical machine translation even human translators often debate about the correctness of a translation. Deviations in the used words, phrase, or word order may be acceptable or have only a weak relation to the acceptance by a human. This directs to the idea to change the objective function to maximize the expected Bleu [He & Deng 12]. The second approach uses similar feature sets, the same RPROP algorithm, and the same exponential probability function. It only changes to the use of maximum expected Bleu and couples more closely to the phrase-based or hierarchical baseline with the use of n-best lists for estimating the derivatives.

5.1 Combination of Hidden Conditional Random Fields with Intermediate Classes

Over the last chapters an effective conditional random fields system was built having high accuracies on natural language understanding and grapheme-to-phoneme conversion. We wanted to evaluate the accuracy of this system on statistical machine translation. However, due to the nature of statistical machine translation with large vocabularies, hidden alignments, reordering, and large training corpora, the application of discriminative methods is only feasible when using effective speed up techniques. We will show that translation models trained with conditional random fields (CRFs) using classes [Goodman 01] are useful in translation, even in addition to a strong baseline. Results with an independent conditional random fields translation system and n-best list rescoring will be presented. To design the tandem of conditional random fields translation model and a phrase-based baseline we will evaluate two different ways of n-best list integrations.

In Section 5.1.1 conditional random fields are summarized together with an extension for hidden variables and a possible implementation, followed by the concept of intermediate word classes in Section 5.1.2, while Section 5.1.3 focuses on the necessary extensions of conditional random fields to be used in statistical machine translation. We discuss the experimental results in Section 5.1.4, and conclude with Section 5.1.5.

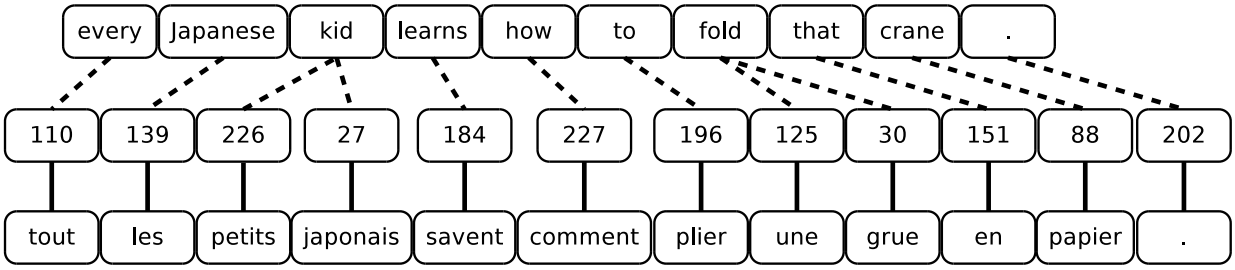


Figure 5.1 Example of decomposition $p(e_1^I|c_1^I, f_1^J)$ / $p(c_1^I|f_1^J)$. First the English source sentence is translated to a class sequence by hidden conditional random fields and finally to the French target sequence by linear chain conditional random fields. The dashed lines mark the alignment with the maximum score of the hidden conditional random fields, while the linear chain conditional random fields have to use a 1-to-1 alignment marked with solid lines. Features are not restricted to the aligned source words and are permitted to use surrounding words as well.

Table 5.1 Two sets of unsupervised word classes estimated by the method described in [Och 95].

number of classes	average # of elements per class	average # of elements per pos.
250	223	162
500	111	66

5.1.1 Conditional Random Fields (CRF)

In this section we will reuse the implementation of linear chain conditional random fields (LCCRFs, Section 3.2.2.1) supporting only 1-to-1 alignments from Section 3.2.4 and the hidden conditional random fields from Section 4.4 used for grapheme-to-phoneme conversion. Both L1- and L2-regularization are used together with a feature set combining the feature sets of attribute name extraction (Section 3.2.2) and grapheme-to-phoneme conversion (Section 4.3). Three types of features $h_l(e_{i-1}, e_i, f_1^J)$ were used to support the conditional probability:

source-n-gram features depending only on one target symbol e_i and a combination of source symbols $f_{A(j)+\gamma_1}^{A(j)+\gamma_2}$ relative to the currently aligned source word $f_{A(j)}$ (with $\gamma_1 \leq \gamma_2$), with $\gamma_1, \gamma_2 = -5, \ldots, 5$, $\gamma_1 + \gamma_2 + 1 \leq 3$,

target-n-gram features describing the relation of a consecutive set of target symbols e_i, e_{i-1}, and

source word stem features, including prefixes and suffixes up to length 4 and capitalization on the aligned source word

As described in Section 3.2.4, the conditional random fields software was implemented with weighted finite state transducers [Mohri 09]. The implementation includes an estimation of the update statistics utilizing a finite state transducer posterior calculation with respect to a log-semi-ring and selecting the best path by utilization of a single source shortest distance (SSSD) operation on a tropical semi-ring.

5.1.2 Word Classes

Maximum Entropy approaches in general and conditional random fields in particular have an unfavorable computational complexity with respect to the size of the used target vocabulary $|\mathbb{E}|$. The computational complexity is a polynomial with a degree equal to the largest target n-gram due to the sums in the denominator in Equation 4.3. E.g. bigrams result in a quadratic complexity.

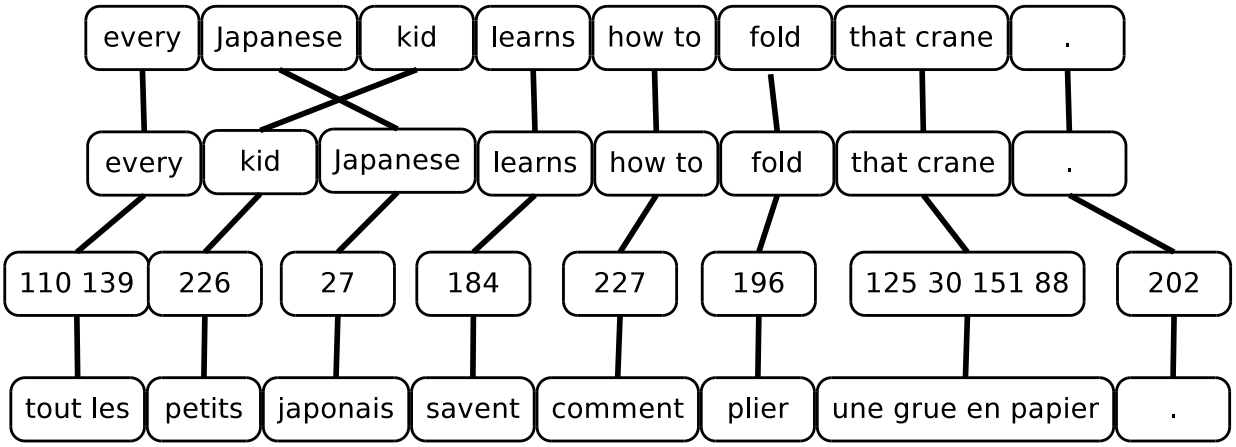


Figure 5.2 Example of the reordering strategy. A phrase is marked with round boxes. The phrase alignment is used to reorder the English source sequence, afterwards the same approach as in Figure 5.1 is applied. In parameter estimation and n-best rescoring the number of target words in a target phrase is restricted to the number of target words in the target phrase as expected by the phrase alignment.

With the help of different optimizations like sparse-forward-backward and pruning in training the computational cost can be shifted to lower polynomial degrees. However, even with these reductions the computation is still very demanding. Our idea is to apply a factorization which already successfully reduced the computational needs of neural networks [Allauzen & Bonneau-Maynard$^+$ 11, Le & Oparin$^+$ 11]. In [Goodman 01], the probability $p(e_1^I|f_1^J)$ is factorized with the help of a clustering function $\gamma : e \mapsto c$, clustering target words e to classes c:

$$\begin{aligned} p(e_1^I|f_1^J) &= \sum_{c_1^I} p(e_1^I|c_1^I, f_1^J)\ p(c_1^I|f_1^J) \\ &\approx \max_{c_1^I} \left\{ p(e_1^I|c_1^I, f_1^J)\ p(c_1^I|f_1^J) \right\} \end{aligned} \tag{5.1}$$

If γ is a strict partitioning clustering, the $\sum_{c_1^I}$ and respectively $\max_{c_1^I}$ can be removed, because $p(e_1^I|c_1^I, f_1^J) = 0$ for all $\gamma(e_i) \neq c_i$. This concept greatly reduces the computational complexity as the effective target vocabulary in $p(c_1^I|f_1^J)$ is the class vocabulary $|C| \ll |\mathbb{E}|$ and the computation in $p(e_1^I|c_1^I, f_1^J)$ can be restricted to only those words e which are part of the already selected class $|\gamma^{-1}(c)| \ll |\mathbb{E}|$. In this section we use the unsupervised maximum entropy word class estimation described in [Och 95] and [Kneser & Ney 93] already used in the preparation of GIZA++ estimations, resulting in two sets of word classes in Table 5.1.

5.1.3 Conditional Random Fields Models for Statistical Machine Translation

Training the conditional random fields (Section 5.1.1) directly for $p(e_1^I|f_1^J)$ with a full translation vocabulary with a size of $55k$ was infeasible. Thus we used word classes (Section 5.1.2) and split the translation process into three steps. The first step models $p(c_1^I|f_1^J)$ with hidden conditional random fields (Equation 4.3). This trained model is used to maximize the alignment A with respect to the reference source f_1^J and target sequences e_1^I. Furthermore, a second model $p(e_1^I|c_1^I, A, f_1^J)$ with linear chain conditional random fields (Equation 3.4) using the given alignment A is estimated (see Figure 5.1). Alignments A in the context of hidden conditional random fields (dashed lines

in Figure 5.1) are only needed to define the position $A(j)$ of source-to-target features $f_{A(j)+\gamma_1}^{A(j)+\gamma_2}$. Features are not restricted to the aligned words and may include surrounding words.

As described in Section 3.2.4 search can be implemented by utilization of a Single Source Shortest Distance (SSSD) operation on the final automaton. With the decomposition into two steps $p(c_1^I|f_1^J)$ and $p(e_1^I|c_1^I, A, f_1^J)$ the search is implemented by consecutive application of SSSD in both automata. We added a LM score c_{LM} and a word penalty c_{WP} by composition to the final automaton

$$-c_{LM} \cdot \log\left(p_{LM}\right) + c_{WP},$$

to compensate that the conditional random fields models have only been trained on the bilingual part of the corpus and some of the conditional random field models lack target-n-gram features (e_{n-1}, e_n). Prior to composition with the LM the word labels *begin* (b), *continue* (c), and *skip* (s) where replaced with the pure word in case of *begin* (b) and *skip* (s) and by an ϵ-label in case of *continue* (c). Composing the LM automaton with the full search space of the (hidden) conditional random fields automata is computationally infeasible, raising the need of posterior pruning before LM composition with a beam pruning threshold of 5. Contrary to phrase-based systems, the resulting models are able to estimate a score $\sum_{i=1}^{I} H(A, e_{i-1}, e_i, f_1^J)$ for any sequence of words, including out-of-vocabulary words (OOVs). In the case of an out of vocabulary word, the features of the conditional random fields with respect to the current word are not activated, but the features from the surrounding region are still activated. However, in statistical machine translation it is often desirable to leave out of vocabulary words untranslated, which are often named entities. So the used software detects out of vocabulary words and translates them with an out of vocabulary word label. There is no further processing of the out of vocabulary word, each out-of-vocabulary word is translated to the same out-of-vocabulary word label.

N-best list rescoring is done by adding scores in addition to the model scores provided by the translation system. A (hidden) conditional random field score $H(a_1^I, c_1^I, f_1^J) = \sum_{i=1}^{I} H(a_i, c_{i-1}^i, f_1^J)$ can be added in multiple ways. Possible implementations are as fully normalized log-probability

$$-\log\left(p(c_1^I|f_1^J)\right) = -\log\left(\sum_{a_1^I} \exp\left(H(a_1^I, c_1^I, f_1^J)\right)\right) + \log\left(\sum_{\tilde{a}_1^I}\sum_{\tilde{c}_1^I} \exp\left(H(a_1^I, c_1^I, f_1^J)\right)\right)$$

or only the numerator of the probability is taken

$$N_{\text{sum}} = -\log\left(\sum_{a_1^I} \exp\left(H(a_1^I, c_1^I, f_1^J)\right)\right) \tag{5.2}$$

or the maximum of the numerator is used

$$N_{\max} = -\log\left(\max_{a_1^I}\left\{\exp\left(H(a_1^I, c_1^I, f_1^J)\right)\right\}\right) = -\max_{a_1^I}\left\{H(a_1^I, c_1^I, f_1^J)\right\}. \tag{5.3}$$

The last variant equates to the maximum approximation in the alignment. In early experiments it turned out that the full normalization did not improve the translation quality, and as the calculation of the full normalization is much more computational demanding, we did not use it.

The (hidden) conditional random fields we used do not support reordering directly, only the features support crossing alignments (e.g. (f_{i-1}, e_i) and (f_i, e_{i-1}) could be used at the same time). To support reordering in parameter estimation we apply forced alignments (FA) [Wuebker & Mauser$^+$ 10] to generate a reference phrase alignment between source and target in the training data (see Figure 5.2). Subsequently, the source sequence was reordered with respect to this phrase alignment (between line one and two in Figure 5.2). Additionally, the phrase alignment fixed the

Table 5.2 IWSLT 2011 evaluation data en→fr

	Train Ted part		Dev 2010		Test 2010	
	En	Fr	En	Fr	En	Fr
sent.	**107k**		**934**		**1.664**	
words	2M	2M	21k	20k	32k	34k
voc.	44k	56k	3.3k	3.8k	3.8k	4.7k
OOVs	-	-	316	392	322	401

Table 5.3 Results with SSSD search within the conditional random fields framework, and independent of any baseline system. The baseline system is only reported for comparison (PBT 1/2). Language model scales and word penalty were selected to optimize Bleu on the dev set.

		word classes	reord-ering	dev 2010 Bleu [%]	dev 2010 Ter [%]	test 2010 Bleu [%]	test 2010 Ter [%]
1	PBT 1 (all data)			28.3	55.9	31.8	50.2
2	PBT 2 (TED)			25.8	58.3	29.4	52.3
3	unigram, with LM	250	no	22.7	60.8	25.7	55.1
4		500	no	22.9	60.7	25.7	55.2
5		250	yes	22.8	60.9	26.6	54.1
6		500	yes	22.8	60.5	26.6	54.1
7	bigram, without LM	250	no	21.6	60.5	24.6	55.1
8		250	yes	21.5	62.3	25.4	55.4
9	bigram, with LM	250	no	21.8	61.3	25.2	55.6
10		250	yes	21.7	61.5	25.1	55.3

number of slots to be filled by the conditional random fields model, i.e. in a phrase with M source and N target words, the conditional random fields model was forced to produce exactly N target words. During search similarly an unconstrained phrase-based translation system was used to generate a phrase sequence. In n-best rescoring the source was reordered as in training with this phrase sequence, applying the same slot constraint. Where in SSSD search only the source was reordered and no restriction on the slots was applied. We have to note that with considering the reordering, some information from a phrase-based translation system is passed to the SSSD search.

Even though the intermediate classes speed up the training, and we also implemented the concept of [Lavergne & Cappé$^+$ 10], a full training of conditional random fields using target bigram features is not possible. To permit the training of conditional random fields with bigram features we included a posterior pruning step in training before applying the bigram features and after the source-n-gram features have been applied. The posterior pruning restricts the number of possible translations in $\sum_{\tilde{e}_1^I, I(A)}$ of Equation 4.3 to a reduced number of summands. As only paths with low scores where removed, most of the probability mass is preserved.

5.1.4 Experimental Results

The experiments were conducted on training and test sets extracted from the English to French data of the International Workshop on Spoken Language Translation (IWSLT) 2012. The (hidden) conditional random field models were trained on the TED part of the training data, and the IWSLT 2012 development and test sets were applied in optimization and evaluation. As baseline system

we were provided with the best single phrase-based translation system from [Peitz & Mansour+ 12] for English to French and a phrase-based translation system with forced alignments only trained on the TED training data. The first phrase-based system used the SCSS software variant of the Jane software package [Wuebker & Huck+ 12] and made use of all available in-domain and out-domain data, part-of-speech-based adjective reordering as preprocessing step [Popović & Ney 04], a language model with all available monolingual training data, and a 7-gram word class language model. The second system is the same system but is solely trained on the TED portion of the training data with the generative training scheme presented in [Wuebker & Mauser+ 10] applying forced alignment. Models include translation probabilities and lexical smoothing in both directions, word and phrase penalty, distance-based reordering model, an n-gram target language model and three binary count features, and a 4-gram language model trained on the TED, Europarl, news-commentary and shuffled news corpora from the workshop. From the shuffled news data 1/8 of the sentences were selected with the technique presented in [Moore & Lewis 10].

5.1.4.1 SSSD Search within the Conditional Random Fields Framework

Table 5.3 contrasts the results of the two phrase-based translation systems (line 1 and line 2) with an independent conditional random field translation tandem $p(c_1^I|f_1^J)$ / $p(e_1^I|c_1^I, f_1^J)$. Line 3 to 6 replaced the bigram features in model $p(c_1^I|f_1^J)$ with a LM estimated on all available monolingual training data in IWSLT projected to classes, while line 7 and line 8 used bigram features instead of a LM, and line 9 and line 10 used both. Results were much less affected by the choice of features in $p(e_1^I|c_1^I, f_1^J)$. LM or bigram features in $p(e_1^I|c_1^I, f_1^J)$ did not result in a change in translation performance. Using reordering improves the generalization of the systems by approximately 1% absolute in BLEU improvement on test except in the case of line 10. With the final translation from classes to the target language $p(c_1^I|f_1^J)$ the gains of the bigram feature and the LM do not add. The translation performance was not competitive to the phrase-based translation systems even when the phrase-based translation was restricted to the same training data (line 2). Phrase-based translation systems are well designed systems with a decade of detailed development, and it could be expected that some aspects of a translation are captured in a phrase-based translation but not in this conditional random fields system, e.g. reordering, why we decided to design a combination of both via n-best rescoring.

5.1.4.2 N-best Rescoring

Both phrase-based translation systems were used to create 1000-best lists. Of all hypotheses with the same resulting word sequence but different phrase segmentation only the one with the best score was added to the n-best list, resulting in a search space with an average of 475 n-best hypotheses per sentence for the development set and 505 n-best hypotheses per sentence for the test set were created with the first phrase-based translation system and 312 n-best hypotheses per sentence for the development set and 329 n-best hypotheses per sentence for the test set with the second phrase-based translation system. To augment these lists, conditional random field models modeling $p(c_1^I|f_1^J)$ and $p(e_1^I|c_1^I, f_1^J)$ for 250 and 500 classes with and without reordering were trained without bigram features. Additionally, two conditional random fields models were trained to capture $p(c_1^I|f_1^J)$ for 250 classes with bigram features. Training conditional random field models with bigrams on 500 classes was still computationally too demanding. These models were used to augment the n-best lists with the scores N_{sum}, and N_{max}. Using fully normalized probabilities did not change the translation quality. On the final augmented n-best lists the weights for n-best list scores were retrained via minimum error rate training (MERT) [Och & Ney 04], initialized with the best weights of the n-best list generating SCSS system.

Experiments have shown that the second model $p(e_1^I|c_1^I, f_1^J)$ did not change the translation

Table 5.4 Results of n-best rescoring adding the (hidden) conditional random fields scores on top of the scores in the n-best lists of the second phrase-based translation system trained on TED data. Line 1 indicates the result of the baseline system. Bold face numbers mark the best result with respect to the dev set.

		word classes	reord-ering	dev 2010 BLEU [%]	dev 2010 TER [%]	test 2010 BLEU [%]	test 2010 TER [%]
1	PBT 2 (TED)			25.8	58.3	29.4	52.3
	(oracle best)			37.3	47.6	43.7	39.7
	(oracle worst)			14.3	75.2	16.3	70.2
2	PBT 2 + N_{sum}	250	no	25.8	58.3	29.4	52.3
3		500	no	26.6	57.5	30.2	51.5
4		250	yes	**26.7**	**57.6**	**29.7**	**51.9**
5		500	yes	25.8	58.3	29.4	52.3
6	PBT 2 + N_{max}	250	no	25.8	58.3	29.4	52.3
7		500	no	**26.5**	**57.6**	**30.0**	**51.4**
8		250	yes	26.5	57.7	29.9	51.6
9		500	yes	25.8	58.3	29.4	52.3
10	PBT 2 + N_{sum}	250	no	26.4	57.7	29.4	51.9
11	(including target bigram)	250	yes	**27.1**	**56.9**	**30.1**	**51.2**

quality, and got a zero weight by the MERT training. The results with only the first model $p(c_1^I|f_1^J)$ are shown in Table 5.4 and Table 5.5. To have a fair comparison the parameters of the baseline system (line 1) were reoptimized, too. We have marked the systems giving the best results with respect to the development set. The best systems on the development set produce the best results on the test set, but in some cases the MERT optimization was not able to include the conditional random fields score in a useful way and we do not gain an improvement in performance. Best improvements in Table 5.4 were +0.3%, +0.6%, +0.7% in BLEU and −0.4%, −0.9%, −1.1% in TER, and +0.4%, +0.3%, +0.4% in BLEU and −0.7%, −0.2%, −0.7% in TER in Table 5.5. The size of improvements were not influenced by the used training data of the phrase-based translation system, thus the improvement is not due to adaptation effects. The results with N_{sum} seem to be a bit more stable than the results with N_{max}, which can be explained by a reduced sensitivity to single bad alignments between the source and target sequence. A gain in using the reordering could not been verified in the n-best experiments.

5.1.5 Conclusion

In this section, we have presented the combination of conditional random fields (CRFs) and intermediate classes, which became popular over the last years in the context of neural network language models, for statistical machine translation. We have shown that intermediate classes give a useful alternative to other speed up techniques already tested by different authors. The technique could e.g. be further extended by a better class selection, and an integrated training of the models $p(e_1^I|c_1^I, f_1^J)$ and $p(c_1^I|f_1^J)$. Additionally we have provided a strategy to include conditional random fields scores with phrase-based translation systems via n-best rescoring. On the unconstrained baseline we can report improvements of up to +0.4% absolute in BLEU and −0.7% absolute in TER.

Table 5.5 Results of n-best rescoring adding the (hidden) conditional random fields scores on top of the scores in the n-best lists of the first and stronger phrase-based translation system. Line 1 indicates the result of the baseline system. Bold face numbers mark the best result with respect to the dev set.

		word class.	re-ord.	dev 2010 BLEU [%]	dev 2010 TER [%]	test 2010 BLEU [%]	test 2010 TER [%]
1	PBT 1			28.3	55.9	31.8	50.2
	(oracle best)			40.8	43.9	46.8	36.7
	(oracle worst)			16.4	72.8	18.4	68.6
2	PBT 1 + N_{sum}	250	yes	28.3	55.9	31.8	50.2
3		500	no	**28.7**	**55.6**	**32.2**	**49.5**
4		250	yes	28.4	55.6	32.0	49.7
5		500	yes	28.5	55.6	32.2	49.5
6	PBT 1 + N_{max}	250	no	28.3	55.9	31.8	50.2
7		500	no	**28.6**	**55.8**	**32.1**	**50.0**
8		250	yes	28.3	55.9	31.8	50.2
9		500	yes	28.3	55.8	31.8	50.2
10	PBT 1 + N_{sum} (including	250	no	**28.8**	**55.3**	**32.2**	**49.5**
11	target bigram)	250	yes	28.3	55.9	31.8	50.2

5.2 Maximum Expected Bleu

Conditional random fields rely on an accurate reference translation, which seldomly can be provided for statistical machine translation. This section reports on a proposal to change to a maximum expected BLEU objective. Both maximum expected BLEU and conditional random fields are a combination of a probability model and a training criterion. They share the same log-linear model but differ in the training criterion. In the experiments of this section, updates are constrained to n-best lists which is similar to the approach of [Blunsom & Cohn$^+$ 08] using a hierarchical system with constrained search space for the calculation of the derivatives. The features are similar to the features used in the last chapters. The only difference is the loss function. Conditional random fields are based on a sentence error loss function while maximum expected BLEU optimize for the BLEU metric. In statistical machine translation the generation of reference translations is hard and agreement between human translators on the reference translation is low. BLEU has a weaker dependency on the correctness of the reference and seems to be the metric of choice to update statistical machine translation systems discriminatively. (As BLEU values are maximized it is strictly speaking a win function.)

5.2.1 Log-linear model combination

To gain a closer understanding of the parameterization of a statistical machine translation system we want to reconsider the log-linear combination (Equation 1.6). Annotating the log-linear combination with model combination parameters $\Lambda = \ldots, \lambda_r, \ldots$ and the parameters of the respective models $\Theta = \ldots, \vartheta_m, \ldots$ gives

$$p_{\Lambda,\Theta}(E|F) = \max_A \{p_{\Lambda,\Theta}(E, A|F)\} \tag{5.4}$$

with

$$p_{\Lambda,\Theta}(E, A|F) = \frac{\exp\left(\sum_r \lambda_r h_{r,\Theta}(E, A, F)\right)}{\sum_{\tilde{E}} \sum_{\tilde{A}} \exp\left(\sum_r \lambda_r h_{r,\Theta}(\tilde{E}, \tilde{A}, F)\right)}. \tag{5.5}$$

Traditional statistical machine translation feature functions are *dense* features activated at each phrase. E.g. the phrase model score from source-to-target is defined as

$$h_{phr}(E, A, F) = \sum_{(\tilde{e},\tilde{f})\in(E,A,F)} \log\left(p_{phr}(\tilde{e}|\tilde{f})\right) \tag{5.6}$$

at the sentence level summed up over the individual components for each phrase $(\tilde{e}, \tilde{f})$ in the hypothesis (E, A, F). In the baseline system, phrase model probabilities are defined as relative frequencies of phrases extracted from word-aligned parallel training data. The joint counts $C(\tilde{e}, \tilde{f})$ of the target phrase $\tilde{e}$ and the source phrase $\tilde{f}$ in the training data are normalized by the marginal counts $C(\cdot)$ of source and target phrase to obtain a conditional probability:

$$p_{phr}(\tilde{e}|\tilde{f}) = \frac{C(\tilde{e}, \tilde{f})}{C(\tilde{f})}. \tag{5.7}$$

The phrase model in the inverse direction is computed in an analog way. Each phrasal log-probability $log\, p_m(\tilde{e}|\tilde{f})$ can be interpreted as a model parameter $\vartheta_{m,(\tilde{e},\tilde{f})} \in \Theta$. Most feature functions $h_{r,\Theta}(E, A, F)$ are composed of logarithmic representation with a sum from smaller components: In phrase models phrases, in lexical models lexical pairs, in language models n-grams. In these features every summand may be interpreted as a model parameter.

Additionally, the $h_{r,\Theta}(E, A, F)$ may include feature functions modeling word or phrase penalties.

5.2.2 Objective Function

Following [He & Deng 12], we want to optimize a maximum expected BLEU objective. The document level BLEU score needs to be approximated with a sentence level BLEU score (Section 5.2.3), which is referenced as β in this section. We denote the space of possible sentences in the source language as $\mathbb{F}^*$ and in the target language as $\mathbb{E}^*$ (* being the Kleene star). The expected BLEU score under parameter set Θ, Λ with respect to the joint probability distribution $p_{\Lambda,\Theta}(\cdot,\cdot)$ is defined as

$$\langle\beta\rangle_{\Lambda,\Theta} = \sum_{F\in\mathbb{F}^*} \sum_{E\in\mathbb{E}^*} p_{\Lambda,\Theta}(E, F)\beta_F(E) = \sum_{F\in\mathbb{F}^*} \sum_{A\in\mathbb{A}^*} \sum_{E\in\mathbb{E}^*} p_{\Lambda,\Theta}(E, A, F)\beta_F(E) \tag{5.8}$$

Here, $\beta_F(E)$ is the BLEU score for target sentence E with respect to the source sentence F (assuming the reference translation to be part of the mapping β) and we use the notation $\langle\cdot\rangle$ to denote the expectation. The joint probability $p_{\Lambda,\Theta}(E, F)$ is decomposed with the help of the Bayes theorem, resulting in:

$$\langle\beta\rangle_{\Lambda,\Theta} = \sum_{F\in\mathbb{F}^*} p(F) \sum_{E\in\mathbb{E}^*} \sum_{A\in\mathbb{A}^*} p_{\Lambda,\Theta}(E, A|F)\beta_F(E)$$

Summing over all possible source and target sentences $\mathbb{F}^*$, $\mathbb{E}^*$ is infeasible. Therefore, we estimate the empirical expectation on a corpus $\{(E_1, F_1), \ldots, (E_k, F_k), \ldots, (E_K, F_K)\}$ of size K. The common used search strategies have constraints on the possible target sequence E with respect to the source sequence F and the applied model Λ, Θ. The target sentences E together with the respective alignment A are then all possible derivations of the source sentence F: $\mathbb{E}_{\Lambda,\Theta}(F)$. The

Bleu score β is measuring the difference between the reference translation E_k and the hypothesis translation E.

$$\langle\beta\rangle_{\Lambda,\Theta} = \frac{1}{K}\sum_{k=1}^{K}\sum_{E,A\in\mathbb{E}_{\Lambda,\Theta}(F_k)} p_{\Lambda,\Theta}(E,A|F_k)\beta(E,E_k)$$

The inner summation over $\mathbb{E}_{\Lambda,\Theta}(F_k)$ can be be approximated by a subset $\tilde{\mathbb{E}}_{\Lambda,\Theta}(F_k)$ of the most likely hypotheses E, A with respect to the parameterized probability $p_{\Lambda,\Theta}(E,A|F_k)$. In practice, we use an n-best list.

$$\langle\beta\rangle_{\Lambda,\Theta} = \frac{1}{K}\sum_{k=1}^{K}\sum_{E,A\in\tilde{\mathbb{E}}_{\Lambda,\Theta}(F_k)} p_{\Lambda,\Theta}(E,A|F_k)\beta(E,E_k) \tag{5.9}$$

The normalized posterior translation probability $p_{\Lambda,\Theta}(E,A|F)$ from source sentence F to target sentence E approximates a maximum entropy model normalized at the sentence level, as used with conditional random fields [Lafferty & McCallum$^+$ 01]:

$$p_{\Lambda,\Theta}(E,A|F) = \frac{e^{\sum_r \lambda_r h_{r,\Theta}(E,A,F)}}{\sum_{E',A'\in\mathbb{E}_{\Lambda,\Theta}(F)} e^{\sum_r \lambda_r h_{r,\Theta}(E',A',F)}}. \tag{5.10}$$

The denominator of this probability does not depend on the output sentence. Thus, the maximization of

$$\operatorname*{argmax}_E \left\{\max_A \{p_{\Lambda,\Theta}(E,A|F)\}\right\} = \operatorname*{argmax}_E \left\{\max_A \left\{\sum_r \lambda_r h_{r,\Theta}(E,A,F)\right\}\right\}$$

is equal to the maximization of the translation score as in Equation 1.7. Here, different from conditional random fields, the expectation of our metric is optimized instead of the conditional log-likelihood. Additionally, conditional random fields are originally normalized over all possible output sequences, without the restriction $E', A' \in \mathbb{E}_{\Lambda,\Theta}(F)$.

Derivative of model parameters Θ

A statistical machine translation system is parameterized with the dense model parameters λ_r per dense model enumerated by r. E.g. a dense model might be the full phrase based model from source to target, the language model, or a re-ordering model. Each dense model r is additionally parameterized with large sets of parameters $\vartheta_{m,r}$. This might be the translation probability of one phrase with the phrase based model, the probability of one n-gram in the language model, or the weight of one re-ordering operation. In the experiments of this section, the dense model parameters are still optimized by the minimum error rate training [Och 03]. Only a set of new model parameters ϑ_m are optimized with the maximum expected Bleu training. When the experiments introduce a set of model parameters (e.g. additional weights per phrase or per re-ordering operation), each set will get a model score λ_r in the minimum error rate training optimization. This model score λ_r is optimized together with the other model scores with minimum error rate training.

To optimize the new set of sparse model parameters Θ, we need to maximize the term $\langle\beta\rangle_{\Lambda,\Theta}$ by setting its derivative to zero.

$$0 = \frac{\delta\langle\beta\rangle_{\Lambda,\Theta}}{\delta\vartheta_{m,r}} = \frac{1}{K}\sum_{k=1}^{K}\sum_{E,A\in\tilde{\mathbb{E}}_{\Lambda,\Theta}(F_k)} \beta(E,E_k)\frac{\delta p_{\Lambda,\Theta}(E,A|F_k)}{\delta\vartheta_{m,r}} \tag{5.11}$$

This depends on the derivative of the probability $p_{\Lambda,\Theta}(E, A|F)$:

$$\frac{\partial p_{\Lambda,\Theta}(E|F)}{\partial \vartheta_{m,r}} = \lambda_r p_{\Lambda,A,\Theta}(E|F) \cdot \left(\frac{\partial h_{r,\Theta}(E, A, F)}{\partial \vartheta_{m,r}} - \sum_{E',A' \in \mathbb{E}_{\Lambda,\Theta}(F)} p_{\Lambda,\Theta}(E', A'|F) \frac{\partial h_{r,\Theta}(E', A', F)}{\partial \vartheta_{m,r}} \right)$$

The feature functions $h_{r,\Theta}(E, A, F)$ are binary features in this work. They are every time activated if a predefined condition happens (e.g. a phrase is used in a hypothesis). Which means that they are equal to the product of the feature weight $\vartheta_{m,r}$ times the number of occurrences $\#_{m,r}$ in a hypothesis. Thus, the derivatives of the feature functions with respect to $\vartheta_{m,r}$ is equal to the number of occurrences $\#_{m,r}$ of a feature m, r in a hypothesis:

$$\frac{\partial h_{r,\Theta}(E, A, F)}{\partial \vartheta_{m,r}} = \#_{m,r}(E, A, F)$$

Putting all this together results in (for ease of notation $\mathbb{E}_{\Lambda,\Theta}(F_k)$ is represented by $\mathbb{E}_k$):

$$\begin{aligned} \frac{\delta \langle \beta \rangle_{\Lambda,\Theta}}{\delta \vartheta_{m,r}} = & \frac{1}{K} \lambda_r \sum_{k=1}^{K} \Big(\sum_{E,A \in \tilde{\mathbb{E}}_k} p_{\Lambda,\Theta}(E, A|F_k) \beta(E, E_k) \#_{m,r}(E, A, F) \\ & - \Big(\sum_{E,A \in \tilde{\mathbb{E}}_k} p_{\Lambda,\Theta}(E, A|F_k) \beta(E, E_k) \Big) \Big(\sum_{E,A \in \tilde{\mathbb{E}}_k} p_{\Lambda,\Theta}(E, A|F_k) \vartheta_{m,r}(E, A, F) \Big) \Big) \end{aligned} \tag{5.12}$$

This can be more compactly expressed by local expectations $\langle \cdot \rangle_k$ and a local covariance cov_k of the BLEU score and the feature count $\#$:

$$\frac{\delta \langle \beta \rangle_{\Lambda,\Theta}}{\delta \vartheta_{m,r}} = \frac{1}{K} \lambda_m \sum_{k=1}^{K} (\langle \beta \#_{m,r} \rangle_k - \langle \beta \rangle_k \langle \#_{m,r} \rangle_k) = \frac{1}{K} \lambda_m \sum_{k=1}^{K} \mathrm{cov}_k(\beta, \#_{m,r}) \tag{5.13}$$

In the implementation, $\#_{m,r}$ is moved to the front of the equation to obtain common factors that can be used by all parameter updates:

$$\frac{\delta \langle \beta \rangle_{\Lambda,\Theta}}{\delta \vartheta_{m,r}} = \frac{1}{K} \sum_{k=1}^{K} \sum_{E,A \in \mathbb{E}_k} \#_{m,r}(E, A, F) \cdot \lambda_m p_{\Lambda,\Theta}(E, A|F) (\beta_k(E, E_k) - \langle \beta \rangle_k) \tag{5.14}$$

Regularization

Maximum Entropy models often generalize better when extended by a regularization term. In [He & Deng 12] the Kullback-Leibler divergence is adapted to be used as regularization. They extended the expected BLEU objective by multiplying with a prior distribution over the parameters $p(\Theta, \Theta^{(0)})$

$$\langle \beta \rangle'_{\Lambda,\Theta} = \langle \beta \rangle_{\Lambda,\Theta} p(\Theta, \Theta^{(0)}) \tag{5.15}$$

and defined this term as

$$p_{KL}(\Theta|\Theta^{(0)}) = C \exp\left(-\tau \sum_{\vartheta \in \Theta} \vartheta \log\left(\frac{\vartheta^{(0)}}{\vartheta} \right) \right).$$

including the hyper parameter τ controlling the degree of regularization. Kullback-Leibler divergence is designed to compare probabilistic distributions, which restricts the possible values of ϑ and $\vartheta^{(0)}$. We apply the more general L2-regularization

$$p_{L2}(\Theta|\Theta^{(0)}) = C \exp\left(-\tau \sum_{\vartheta \in \Theta} (\vartheta - \vartheta^{(0)})^2 \right), \tag{5.16}$$

which has no restrictions on the parameters. Deriving $\langle\beta\rangle'_{\Lambda,\Theta}$ would mix the derivation of the regularization and the non-regularized term. The result of the maximization does not change if done in the log domain and so we optimize the objective function $O(\Theta|\Lambda)$:

$$O(\Theta|\Lambda) = \log\left(\langle\beta\rangle'_{\Lambda,\Theta}\right) = \log\left(\langle\beta\rangle_{\Lambda,\Theta}\right) - \tau\sum_{\vartheta\in\Theta}(\vartheta - \vartheta^{(0)})^2. \tag{5.17}$$

In our experiments we left the existing model parameters $\vartheta_{m,r}$ unchanged. We added new feature sets $\vartheta_{m,r}$ with an own dense model weight λ_r with r larger than the existing number of dense models. The new feature sets $\vartheta_{m,r}$ were initialized with zero and the λ_r to one. The prior model $\Theta^{(0)}$ in this case is just the zero vector $\Theta^{(0)} = \ldots, 0, \ldots$. The derivative of the objective function for the model parameters $\vartheta_{m,r}$ (the λ_r's are tuned with minimum error rate training), follows as:

$$0 = \frac{\delta O(\Theta|\Lambda)}{\delta\vartheta_{m,r}} = \frac{1}{\langle\beta\rangle_{\Lambda,\Theta}}\cdot\frac{\delta\langle\beta\rangle_{\Lambda,\Theta}}{\delta\vartheta_{m,r}} - 2\tau\vartheta_{m,r} \tag{5.18}$$

with $\frac{\delta\langle\beta\rangle_{\Lambda,\Theta}}{\delta\vartheta_{m,r}}$ from the last section. This equation needs to be solved numerically, which is done in our experiments with the RPROP training [Riedmiller & Braun 93].

5.2.3 Sentence-level Bleu-4

Maximum expected BLEU is designed to optimize a sentence-level error metric $\beta_n(E)$. However, the regular BLEU metric (cf. Section 1.3.4) is defined at the document level. The n-gram counts and the brevity penalty are calculated over the full evaluation corpus. In [He & Deng 12], the authors propose to use the smoothed and unclipped sentence-level BLEU-4 score, which we denote as $\beta_n(E) = \text{BLEU}(E, \hat{E}_n)$ with respect to the reference translation $\hat{E}_n$. It is defined as

$$\beta_n(E) = BP(E)\cdot\exp\left(\frac{1}{4}\sum_{n=1}^{4}\log\left(p_n\right)\right) \tag{5.19}$$

with smoothed precisions

$$p_n = \frac{C(\text{n-grams matched}) + \eta\cdot p_{n-1}\frac{p_{n-1}}{p_{n-2}}}{C(\text{n-grams}) + \eta}. \tag{5.20}$$

For $n \in \{3, 4\}$ a smoothing factor of $\eta = 5$, and for lower order n relative frequencies are used ($\eta = 0$). The brevity penalty BP is approximated with a non-clipped penalty

$$BP(E) = \exp\left(1 - \frac{|E|}{|\text{1-best}(\mathbb{E}_n)|}\right) \tag{5.21}$$

subject to the hypothesis length $|E|$ normalized to the baseline hypothesis length $|\text{1-best}(\mathbb{E}_n)|$. In [He & Deng 12] this is motivated by a utility function emphasizing correct n-grams and not hypothesis length.

5.2.4 Leave-one-out

Within these experiments we apply the phrase-based model estimated on the training corpus. Not only the probabilities are extracted from the same corpus. Also the phrase pairs are extracted from the same corpus. If the phrase penalty is positive the phrase alignment has least cost when it is composed with the lowest number of phrases. This is achieved by long spanning phrases which does not match the situation of regular decoding on a corpus separate from the training corpus.

In a normal decoding, the unigram and bigram phrases dominate the alignment. In [Wuebker & Mauser$^+$ 10], the authors propose to compensate the tendency to select unusually long phrases by reducing the counts in the phrase probability (Equation (5.7)) by the count in the specific sentence pair.

We propose to use leave-one-out during the estimation of the n-best lists $\mathbb{E}_k$. Without leave-one-out we expect that the n-best lists would be dominated by unrealistically long phrases. In [He & Deng 12], the authors claim that the n-best lists are sufficient without leave-one-out. However, the experiments in this section show a 0.5% improvement in BLEU by using leaving-one-out.

5.2.5 Features

Maximum expected BLEU can optimize any type of features which can be extracted from the n-best lists $\mathbb{E}_k$. Experiments are performed with four types of features:

phrase features, i.e. one for each phrase pair in the phrase table

lexical features, i.e. one feature for each source-target word pair within the same phrase pair

triplet features on source and target [Hasan & Ganitkevitch$^+$ 08], i.e. one feature for one source and two target words or two source and one target word within the same phrase pair

reordering features [Galley & Manning 08, Cherry & Moore$^+$ 12, Cherry 13], i.e. one feature for each combination of phrase pair, orientation (monotone, swap, or discontinuous), and direction (forward or backward)

Experiments will compare the different update strategies. Experiments using the growth transformation strategy (GT) of [He & Deng 12] will stick to the phrase feature set. Therefore, we will perform a comparative experiment with a set of update strategies using only the phrase feature set. In the phrase-based system the feature types are arranged as dense features $h_{r,\Theta}(E, F)$ in the log-linear model combination (Equation 5.4) with a combination parameter λ_r. The phrase and the lexical features set share one combination parameter, the triplet features are weighted with one parameter for the source and one for the target triplet features, and the hierarchical reordering features have six weights for all combinations of the orientation and the direction.

5.2.6 Final Training Procedure

As the optimization described in this section depends on high quality n-best lists, the method is best suited to discriminatively optimize a given baseline. The algorithm is outlined in Figure 5.3. The component weights Λ of the baseline system are optimized with the MERT algorithm. This baseline is used to generate the n-best lists which are augmented with the sentence-level BLEU values. Now, maximum expected BLEU updates the single features $\vartheta \in \Theta$ for a fixed number of iterations $1, \ldots, t, \ldots, T$ resulting in a set of parameters per iteration Θ_t. The model combination weights Λ are not updated during this iterations. A final run of the MERT algorithm adjusts the model combination parameters for each iteration Λ_t and the best pair Θ_t, Λ_t is selected on a development set.

5.2.7 Experimental Results

These experiments are designed to answer the following five questions:

1. Does the maximum expected BLEU improve a baseline system?
2. Is the RPROP-algorithm the right choice?
3. Do the additional features (lexical, triplet, and reordering) help?
4. Does leaving-one-out improve results?
5. Does the final system exceed the best discriminative baseline?

This questions are answered on three translation tasks (IWSLT de→en, BOLT zh→en, WMT de→en) with three different baseline systems.

Input: baseline system with MERT optimised Λ
Output: discriminative feature parameters Θ and updated log-linear combination Λ_0
  for all training samples F_k, E_k **do**
    Generate n-best list $\mathbb{E}_k$
    for all training samples $E \in \mathbb{E}_k$ **do**
      Compute sentence-level BLEU $\beta(E, E_k)$
    end for
  end for
  $\vartheta_{1,1}^{M,R} = \Theta \leftarrow 0$
  for t=0 **to** T **do**
    for m=0 **to** M **do**
      for r=0 **to** R **do**
        $\frac{\delta O(\Theta)}{\delta \vartheta_{m,r}} \leftarrow -\tau \cdot 2\vartheta_{m,r} + \frac{1}{\langle\beta\rangle_{\Lambda,\Theta}} \cdot \frac{\delta \langle\beta\rangle_{\Lambda,\Theta}}{\delta \vartheta_{m,r}}$
      end for
    end for
    $\Theta_t \leftarrow RPROP(\Theta_t, \Theta_{t-1})$
  end for
  for t=0 **to** T **do**
    $\Lambda_t \leftarrow MERT(\Lambda_0)$
  end for
  Select best $\Theta^{(t)}$ on DEV
  Evaluate on test sets

Figure 5.3 The maximum expected BLEU training algorithm.

Table 5.6 Statistics for the bilingual training data of the IWSLT 2013 German→English, the DARPA BOLT Chinese→English, and the WMT 2014 German→English tasks.

	IWSLT		BOLT		WMT	
	German	**English**	**Chinese**	**English**	**German**	**English**
Sentences	138K		4.08M		4.09M	
Run. Words	2.63M	2.70M	78.3M	85.9M	105M	104M
Vocabulary	75.4K	50.2K	384K	817K	659K	649K

Baseline Setups

The task of the IWSLT 2013 German→English evaluation provides an in-domain bi-lingual TED corpus. This corpus is the best fitting corpus to the test conditions. Its limited size (138k sentences, Table 5.6) makes it an excellent candidate to develop and test computationally expensive discriminative training methods. This TED corpus is used for the generation of the translation table and the maximum expected BLEU training. Additionally, the baseline system is composed of a 4-gram language model, a hierarchical reordering model [Galley & Manning 08], and a 7-gram word class language model [Wuebker & Peitz$^+$ 13]. The language model is estimated with modified Kneser-Ney smoothing from the TED corpus, the news commentary, Europarl v7, common crawl, and a selected part of the shuffled news and LDC English Gigaword corpora. The selection is based on cross-entropy difference [Moore & Lewis 10]. MERT includes random restarts resulting in a non-deterministic behavior. On IWSLT, all results are averages over three independent MERT runs, and we evaluate statistical significance with *MultEval* [Clark & Dyer$^+$ 11].

To confirm the findings we applied maximum expected BLEU on two strong baselines includ-

Table 5.7 Comparison of different update strategies for maximum expected BLEU on the IWSLT 2013 German→English task in BLEU [%]. The comparison between update strategies is done with only the phrase feature type and *RPROP all features* additionally uses lexical, triplet, and reordering features. GT, SGD, AdaGrad and RPROP are trained with leave-one-out, unless otherwise specified.

IWSLT de-en	**# features**	**test** BLEU [%]
baseline	18	30.4
GT [He & Deng 12]	6.08M	30.9
SGD [Auli & Galley+ 14]	921K	30.8
AdaGrad [Green & Wang+ 13]	921K	31.1
RPROP (this work)	921K	**31.3**
RPROP w/o leave-one-out	921K	30.7
RPROP all features	5.22M	**31.6**

ing recurrent neural language models trained on two large-scale tasks. On the DARPA BOLT Chinese→English we use a hierarchical phrase-based SMT engine with 19 dense features including an LSTM recurrent neural network language model in rescoring [Sundermeyer & Schlüter+ 12], a 5-gram backoff language model, and a hierarchical reordering model [Huck & Wuebker+ 13]. The 5-gram language model is estimated on 2.9 billion running words. The discussion forum portion of the training data (67.8k sentence pairs) is used for the maximum expected BLEU training. For tuning we used a set of 1275 sentences and testing is performed on a single-reference set of 1124 sentence pairs taken from the discussion forum data, a web data set of 1239 sentences taken from LDC2010E30, and the NIST MT06 evaluation set. The tuning and web data test set are the same as in [Setiawan & Zhou 13].

Both other systems used only a limited set for the maximum expected BLEU training. The final system developed on the German→English task of the 9th Workshop on statistical machine translation (WMT, [Bojar & Buck+ 14]) uses the full bilingual training corpus for maximum expected BLEU training. The baseline system is a non-hierarchical phrase-based translation system composed of a 4-gram backoff language model estimated from 2.5 billion running words, a recurrent neural language model, a 7-gram word class language model and a hierarchical reordering model.

Table 5.6 summarize the used bilingual corpora.

IWSLT results

Results on the IWSLT data are shown in Table 5.7. The central block in this table makes use of only the phrase feature type to have a fair comparison to the growth transformation algorithm (GT, [He & Deng 12]). All features are added in the final row. The first question if maximum expected BLEU can improve a baseline system is clearly answered with absolute improvements in the range of 0.3% up to 1.2% BLEU. Already the maximum expected BLEU training with only phrase features outperforms the baseline by absolute 0.9% BLEU at a significance level of 99%. Leaving-one-out has a clear impact of 0.5 absolute in BLEU and adding the lexical, triplet, and reordering features gives an additional boost of absolute 0.3% in BLEU.

Extensive experiments have been carried out to verify that the simple RPROP-algorithm is able to outperform the growth transformation (GT), the stochastic gradient descent (SGD), and AdaGrad update strategies. In the comparison, all update strategies use n-best lists generated with leaving-one-out. For all setups the regularization and for SGD and AdaGrad the learning

Table 5.8 Comparison of different update strategies for maximum expected BLEU on the BOLT Chinese→English task in BLEU [%] on the discussion forum test set, the mixed web test set and NIST MT06. The baseline is our BOLT evaluation system and contains a recurrent neural LM. We compare with [Setiawan & Zhou 13] who applied maximum expected BLEU training with growth transformation (GT). Note that the number of features reported in [Setiawan & Zhou 13] is artificially blown up due to renormalization.

BOLT zh-en	**# features**	**discussion forum** BLEU [%]	**web** BLEU [%]	**MT06** BLEU [%]
baseline	-	18.0	34.1	39.7
SGD	12.4M	18.0	34.3	39.8
AdaGrad	12.4M	18.3	34.7	40.1
RPROP	12.4M	**18.7**	**34.8**	**40.5**
Setiawan&Zhou (GT)	150M	-	32.7	40.3

rate were selected on a validation set (test2011). Regularization had only little effect on the error rates. Only a small fixed regularization value is needed. Error rates on a held out set converge after 5-10 iterations for GT and AdaGrad while RPROP and SGD need around 25 iterations. However, while RPROP, SGD, and AdaGrad update independent real valued features without the need of normalization, the growth transformations update directly the phrase probabilities. This raises the need of re-normalizing the phrase probabilities which comes at the cost of a higher memory consumption and longer computation time. RPROP, AdaGrad, and SGD only use one third of the memory of GT ($2.1GB$ / $6.7GB$) and the computation is faster with 2.5 hours for 40 iterations compared to 16 hours for GT. In terms of error rates RPROP outperforms all other update strategies. The smallest difference of absolute 0.2% BLEU between RPROP and AdaGrad is statistically significant at the 95% level.

BOLT results

In Table 5.8 we verify the comparison of the different update strategies. SGD, AdaGrad, and RPROP now train the same strong baseline with the same feature set of lexical, phrase, and triplet features. As the test set had been identical the last row reports directly the results reported by [Setiawan & Zhou 13] using the growth transformation (GT) algorithm. Compared to the individual baselines RPROP gains nearly twice the absolute improvement as reported in [Setiawan & Zhou 13] (+0.7% / +0.44% BLEU) on the *web* data. With an improvement of absolute +0.8% BLEU on MT06 the RPROP is better than the results reported in [Setiawan & Zhou 13] by absolute 0.2% BLEU. The maximum expected BLEU is trained on discussion forum data. However, the gain on the discussion forum test data is similar to the gains on the other test corpora convincing that the improvement is not subject to domain adaptation effects.

Comparing the different update strategies, RPROP clearly outperforms SGD which has only minor improvements over the baseline system. AdaGrad has similar improvements as RPROP but is still absolute 0.1% - 0.4% BLEU worse.

WMT results

The final experiment shows a system where maximum expected BLEU was applied to the full bilingual training corpus of 4M sentence pairs. The generation of the n-best lists took about 3/4 of a month followed by about a week of maximum expected BLEU training. Lexical, phrase, and

Table 5.9 Improvement of a maximum expected BLEU system to its baseline without maximum expected BLEU training on the WMT German→English task in BLEU [%]. The baseline contains a recurrent neural LM.

WMT de-en	**# features**	**newstest2013** BLEU [%]
baseline	-	28.3
+ maximum expected BLEU	45.0M	**28.9**

reordering features are used, summing up to 45M features. On newstest2013 we can report an improvement of 0.6% BLEU.

Summary

Using maximum expected BLEU and RPROP as update strategy improve over a range of tasks strong baselines. Additional features as lexical, triplet, and reordering features can help as well as leaving-one-out. On the BOLT and WMT data sets, improvements in BLEU compared to other published results could be presented. On IWSLT, our method shares the best result with other optimization strategies.

5.3 Conclusion

Among the different tasks maximum entropy models have been applied, statistical machine translation is one of the hardest. The input *and* the output are both from large vocabularies. The alignment is complex and may correspond to long spanning dependencies. Phrase translations are needed. This results in a huge set of possible translations of a source sentence and even leads to a low agreement rate between human translators about the right translation. The estimation of maximum entropy models cannot visit the full hypothesis space of all translations. As possible solutions we present intermediate classes and estimations on n-best lists. To compensate the uncertainty in the reference translation maximum expected BLEU is applied.

The scientific contributions of this chapter have already been published in the conference proceeding [Lehnen & Peter$^+$ 13, Wuebker & Muehr$^+$ 15].

The work presented in this chapter was the work of multiple authors. The contributions of the different authors can be summarized as:

- Combination of Hidden Conditional Random Fields with Intermediate Classes
  The idea and the modifications to the HCRF software came from *Lehnen*. He was supported by n-best lists from *Peter*, the baseline SMT system and forced alignments from *Wübker*, and the second baseline SMT system by *Peitz*.
- Maximum Expected BLEU
  Wübker had the idea for these experiments and modified the phrase model training for them. *Lehnen* proposed to change the algorithm to use RPROP, L2-regularization, and not to update the original models but to add the features as additional models. The first implementation and experiments were mainly done by *Mühr* and extended by *Wübker*.

The authors were part of RWTH Aachen University during the mentioned research.

6. Conclusion

In the first chapter of this work, we reviewed a set of machine learning methods with the result that the discriminatively trained and sentence normalized conditional random fields show a clearly better performance than all other approaches. The difference even got larger when applied on noisy input data (from automatic speech recognition) or extending the objective function with a margin extension. Applying the same learning method on a later step in the natural language understanding pipeline shows improvements, too.

We concluded that this method must be beneficial on other tasks as grapheme-to-phoneme conversion and statistical machine translation. However, these tasks have the need to include structure modelling. Hidden conditional random fields were implemented to extend the conditional random fields with a latent variable. The proposed method of using a simple HMM-like symbol alignment (characters for grapheme-to-phoneme conversion, words for statistical machine translation) together with highly-overlapping features including joint-n-grams has shown to be very effective for grapheme-to-phoneme conversion. The word alignment permits a huge search space increasing the computation time and the highly overlapping features let the memory exceed on our computing environment. So we included different speed up and memory reductions technics.

Even after applying all speed-up technics, the computation time of the training of the hidden conditional random fields was still prohibitive on statistical machine translation. Therefore, we introduced an intermediate modelling layer with predefined word classes, which reduced the training time to an acceptable time span and resulted in translation performance improvements. Optimizing the shape of the model and its training criterion has shown effective on natural language understanding (maximum entropy Markov models vs. conditional random fields; the margin extension) so we tried maximum expected Bleu training. Which is similar to conditional random fields but optimises the Bleu metric. To address the computation time constraints we trained these models on top of existing statistical machine translation systems. In this setup, we needed to model the structure with the baseline system, which was a phrase-based system. The maximum expected Bleu training on top of a baseline could improve consistently all baseline results.

7. Outlook

We have seen that maximum entropy training can have higher performance than other generative and discriminatively trained systems but the objective function and the modelling of the structure between source and target sequence have to be carefully chosen. On grapheme-to-phoneme conversion, we have shown that features using the modelled structure only as offsets but ignoring the alignment borders works very well. In recent literature we see the emerge of neural machine translation. Neural machine translation improved the first time over a strong baseline [Bahdanau & Cho$^+$ 15] when combined with a weak alignment, which is a similar finding as in our experiment. This seems to be a promising line of research for all discriminative learning models.

Running the experiments for this thesis has taken a huge amount of computational resources. Applying technics from this thesis in an industrial setting works for natural language understanding but especially statistical machine translation would need much more speed-up. Neural network training became attractive at the moment accelerators could be applied. However, conditional random fields commonly use large sets of binary sparse features while the accelerators are designed for dense memory structures. A solution to this computational problem would speed up the development of maximum entropy models a lot.

Our experiments have shown that maximum expected Bleu can improve existing statistical machine translation systems. Neural machine translation uses a final soft-max layer. A soft-max layer has a very similar structure as the probability optimized in the maximum expected Bleu section of this thesis. Could the soft-max layer be trained with maximum expected Bleu?

8. Scientific Contributions

This thesis is organized around maximum entropy sequence models with an emphasis on conditional random fields (CRFs). Work was carried out over three different tasks. First semantic tagging within natural language understanding (NLU) with an input vocabulary of natural language (size 1k-100k) and an artificial output vocabulary (size 100-300) were considered. Segmentations are generally provided within semantic tagging. Within the second task, grapheme-to-phoneme conversion (G2P), the input and output label sets are letters and phonemes (size 26-50). The additional complication are a missing segmentation and very complex feature sets. Finally, statistical machine translation (SMT) with natural language input and output, missing alignments, and complex feature sets has been tackled.

A detailed analysis and comparison of conditional random fields on three corpora (NLU) Beginning the task of semantic tagging within natural language understanding, it is a critical question, which methods to select to build a semantic tagging system. We set up several approaches and in the end implemented a maximum entropy software package. A detailed comparison of the different approaches on reference and automatic speech recognition input on the three different corpora from three different languages and domains are presented. Including the margin extension (see below), the best result in terms of concept error rate of conditional random fields on the evaluation part of the MEDIA corpus is 10.6% on reference and the next best system are support vector machines with 13.4%.

Design of a multi-level constrained conditional random fields search (NLU) Within the corpora understudy in the natural language understanding part of the thesis a decomposition of the task into two levels was possible. The level of attribute name extraction assigned a category to one or many words in an input sequence, e.g. "bus number", "hotel name", or "response". In a second level, the attribute value extraction maps the input words within a attribute name to a normalized value. For bus numbers, these may be natural numbers, hotel names are a dictionary set of hotel names seen in the corpus, and a response may be "yes" or "no". We designed an extension of conditional random fields able to respect the constraints provided by a attribute name, including to generate only one label for multiple words and respecting the set of possible output labels. E.g. producing a "yes" as bus number is prohibited.

Combination of conditional random field attribute value extraction with rule based approaches (NLU) It turned out that there are attribute names were the constrained conditional random fields generate high accurate results, while with other attribute names hand crafted normalization scripts are superior. E.g. the words within the attribute name "response" may be unnormalized by nature ("ah, ok", "yes", "mhm", "I dislike that choice"), which makes it hard to create hand crafted rules, but can be mapped within a statistical approach like conditional random fields. Another example would be the attribute name "bus number", which can reach high numbers in the range

of some hundreds, but are easy to process with hand crafted rules. We designed an automatic optimization method and the adapters to the statistical and rule based approach to choose the optimal method for each attribute name. On the MEDIA corpus the result for a combination of the rule based approach with conditional random fields is 12.6% on reference input of the evaluation set compared to 13.5% with the rule based approach.

Testing of the margin extension (NLU) In [Heigold & Schlüter$^+$ 09], Georg Heigold introduced the large margin extension of the maximum mutual information training in automatic speech recognition. Maximum mutual information training is equivalent to conditional random field training and the extension can be also used in our software suite. We tested the tagging performance gain of this method and discovered an improvement with respect to the concept error rate.

An efficient hidden conditional random field implementation adopting the BIO scheme (G2P) In semantic tagging, the corpora are usually equipped not only with a source sentence and a reference target sequence, but additionally with a reference segmentation of the reference target sequence with respect to the source sequence. In many tasks, including grapheme-to-phoneme conversion and statistical machine translation, a segmentation or respectively an alignment is rarely provided with the corpus. Within grapheme-to-phoneme conversion, the second task considered in this thesis, we extended the conditional random field software used within natural language understanding to include the segmentation as a hidden variable. This approach was superior to splitting the process to a segmentation phase using external methods and a labeling phase using conditional random fields. The hidden conditional random fields lead to a word error rate (sequence error rate) of 11.6% on the evaluation set on the CELEX corpus split shared with [Bisani & Ney 08] compared to 12.1% but using an external alignment.

Elastic-Net implemented within RPROP (G2P) Even though the vocabularies are only in the range of 26 to 50 labels within grapheme-to-phoneme conversion, the models can become huge, because of the used source-n-grams and joint-n-grams. Source-n-grams model combinations of multiple source labels together with one target label, and joint-n-grams model combinations of multiple source labels with multiple target labels. We extended the well known RPROP Algorithm to support the L1 regularization in conjunction with the already used L2 regularization. The L1 regularization introduce a jump discontinuity in the gradient, which needs to be addressed in a gradient based optimization algorithm. With this extension sparsity (most feature weights become zero) can be reached. This sparsity is used with efficient data structures to reduce the memory consumption of the software. This extension of the optimization algorithm is general and can be applied to any feature function, not only to source- or joint-n-grams. In a typical setup of grapheme-to-phoneme conversion the number of features can be reduced by a factor of 20.

Sparse-forward-backward for arbitrary n-gram sizes (G2P) Original conditional random fields use all possible target sequences for normalization including sequences which are never seen in the training corpus. However, one can reduce the set of sequences to those sequences which can be constructed from the seen n-grams in the target side of the training corpus. n is chosen to be equal to the longest target-n-gram feature. In the lecture [Ney 09], p. 116 or in the publication [Lavergne & Cappé$^+$ 10] a reduction of the computation cost for target-bigrams is suggested. In this thesis, this reduction was extended to arbitrary n-grams and implemented in a way similar to the backing-off in language modeling (cf. e.g. [Chen & Goodman 99]). Computation time improvements are mainly present when using n-grams longer than bigrams.

Beam pruning in hidden conditional random field training (G2P) During search, the application of a combination of dynamic programming and beam pruning (e.g. [Ney & Mergel$^+$ 87]) is a fairly common approach. However, we had not seen that this strategy is often used in parameter estimation, despite the potential of this method is high for parameter estimation, and in grapheme-to-phoneme conversion there exists the possibility to observe the switchover from full normalization to beam pruned normalization. We implemented and reviewed beam pruning in parameter estimation for conditional random field with a significant speed up in training time. With only target-bigrams, the training time improvement is about 2-3, while with target-bigrams together with target-trigrams the training time improvement is about a factor of 10.

Analysis of the influence of joint-n-grams (G2P) In [Jiampojamarn & Cherry$^+$ 10], it was observed that joint-n-gram features, a combination of multiple source labels together with multiple target labels, are critical for a high accuracy. However, a huge variety of joint-n-gram features are in every grapheme-to-phoneme training corpus. Thus, effective feature selection techniques are needed to select a proper set of features fitting in the memory of the estimation process. We make use of the Elastic-Net feature selection in combination with feature cut-offs to get this set. On the CELEX corpus, joint-n-grams lead to a word error rate (sequence error rate) of 11.0% on the evaluation set compared to 11.7% without joint-n-grams. With joint-n-grams the approach with hidden conditional random fields has an improved word error rate compared to the still best generative approach [Bisani & Ney 08], with a word error rate of 11.4% on the same set.

Analysis of the search space (G2P) Taking all achievements within grapheme-to-phoneme conversion together results in a system with very high accuracy. Unfortunately, even with beam-pruning and sparse-forward-backward the computation time is still very high. In a joint work with partners from LIMSI, our approach was compared in detail to their conditional random fields approach using phrase segmentations assisted by the n-gram based approach [Casacuberta & Vidal 04, Mariño & Banchs$^+$ 06]. It turned out that our approach has higher accuracy (11.7% without joint-n-grams, compared to 14.0%), but the other approach is faster by a factor of thousand.

Combination of hidden conditional random fields with intermediate classes (SMT) Objective of this contribution was to test our hidden conditional random field approach designed during the work on natural language understanding and grapheme-to-phoneme conversion on statistical machine translation. As the training time was still high for this method, we adopted the idea of intermediate classes [Goodman 01] for sequences. The used hidden conditional random field software supports non-monotonous alignments only within the features, which are allowed to cross $((e_i, f_{j-1}), (e_{i-1}, f_j))$, but not with the direct word alignment. In the experiments both the direct approach and an approach reordering the source sentence via forced alignments was tested. Improvements from 31.8% to 32.2% in BLEU and 50.2% to 49.5% in TER are reached with n-best rescoring on the 2010 test set of the IWSLT English to French corpus.

Maximum expected Bleu training (SMT) Conditional random fields with intermediate classes could improve the performance of a statistical machine translation system. However, this improvement is not in the range as expected from other tasks like natural language understanding or automatic speech recognition. Conditional random fields minimize a sentence level $\{0, 1\}$ loss function. It turned out that this loss function is too strong. Changing to a maximization of a sentence-level BLEU results in the expected improvements. E.g., a change from 30.4% to 31.6% in BLEU on IWSLT de→en 2013.

9. Pre-Publications and Joint Work / Individual Contributions vs. Team Work

The work presented in this thesis was done in cooperation with various authors. Here, we explicitly state the individual contribution of the author of this thesis in relation to the other authors based on the list of pre-publications. Additionally, the three conclusion sections in the main body of this thesis (Sections 3.4, 4.9, 5.3) state the individual contributions ordered with respect of the outline of this thesis. If not stated otherwise the authors were part of RWTH Aachen University during the mentioned research.

[Hahn & Lehnen$^+$ 08a], [Hahn & Lehnen$^+$ 08b] In these first two comparisons of different machine learning methods on the task of attribute name extraction, S. Hahn and C. Raymond[1] prepared the corpus, P. Lehnen performed the conditional random fields and maximum entropy Markov models (called *log-pos* in this paper) experiments, C. Raymond[1] performed the finite state transducers and support vector machines experiments, and S. Hahn prepared the statistical machine translation results.

[Heigold & Lehnen$^+$ 08, Heigold & Wiesler$^+$ 10] These two publications are not part of this thesis. They are about the differences of discriminative HMMs, log-linear models, and CRFs. The work was lead by G. Heigold. P. Lehnen helped in verifying the derivations analytically and by conducting experiments.

[Lehnen & Hahn$^+$ 09] In this joint work with A. Mykowiecka[2], A. Mykowiecka[2] designed the Polish NLU corpus including the ontology and the annotation specification. P. Lehnen and S. Hahn pre-processed the corpus and applied their conditional random field toolkit to it. The constrained conditional random field search developed by P. Lehnen is first applied on attribute value extraction in this paper.

[Hahn & Lehnen$^+$ 09] S. Hahn and P. Lehnen applied the margin extension implemented by G. Heigold to the natural language understanding tasks.

[Hahn & Dinarelli$^+$ 11] This journal publication concluded the final results of the LUNA project on attribute name and value extraction on the three different corpora prepared in the project. S. Hahn was the lead for this work. He prepared everyone with pre-processed corpora, collected the results and performed the system combination.

The CRF and MEMM experiments on attribute name extraction were done by S. Hahn and P. Lehnen with the software developed by P. Lehnen. The phrase-based translation results

[1] Christian Raymond, University of Avignon, now Univ. IRISA-INSA
[2] Agnieszka Mykowiecka, Polish Academy of Sciences and Polish-Japanese Institute of Information Technology
[3] Marco Dinarelli, University of Trento, now CNRS-LaTTiCe
[4] Fabrice Lefèvre, Université d'Avignon et des Pays de Vaucluse
[5] Renato De Mori, Université d'Avignon et des Pays de Vaucluse, now McGill University
[6] Giuseppe Riccardi, University of Trento
[7] Alexandre Allauzen, Thomas Lavergne, Francois Yvon, Univ. Paris-Sud, France and LIMSI/CNRS - Spoken Language Processing group

were prepared by S. Hahn and D. Vilar, while the support vector machines results came from M. Dinarelli[3], the finite state transducers results from M. Dinarelli[3] and C. Raymond[1]. The dynamic Bayesian networks results were prepared by F. Lefevre[4].
The rule based attribute value extraction was developed and the results were provided by M. Dinarelli[3] and C. Raymond[1]. The combination of this rule based approach with a statistical approach was done by P. Lehnen and S. Hahn. The DBN approach used for comparisons was developed and the results were provided by F. Lefevre[4].
The introduction to this paper was partly provided by R. de Mori[5]. He supervised the work and the writing of this paper together with H. Ney and G. Riccardi[6].

[Huck & Ratajczak$^+$ 10] This publication is not part of this thesis. This work was published by M. Huck. P. Lehnen helped in preparing the text of the paper.

[Lehnen & Hahn$^+$ 11b] The idea to use sparse-forward-backward and equations on bigram level were first done by H. Ney and T. Lavergne[7]. The algorithm and the implementation for arbitrary n-grams was done by P. Lehnen.

[Heigold & Hahn$^+$ 11] G. Heigold proposed a first solution based on the EM algorithm to the alignment problem of conditional random fields on grapheme-to-phoneme conversion. S. Hahn and P. Lehnen helped with setting up experiments and analysing results.

[Lehnen & Hahn$^+$ 11a] Here, the EM algorithm is extended and compared to a first hidden conditional random fields solution. The EM solution was extended mainly by S. Hahn while the hidden conditional random fields solution was lead by P. Lehnen. A. Guta supported in implementing extensions to the conditional random field software and conducting experiments.

[Hahn & Lehnen$^+$ 11] In this publication three extensions to conditional random fields are presented. The elasic-net extension to the RPROP algorithm designed by P. Lehnen, the integration of a language model into conditional random fields developed by S. Hahn, and the margin extension developed by G. Heigold. The software implementation and experiments were done together by P. Lehnen and S. Hahn.

[Rybach & Hahn$^+$ 11] The conditional random fields software presented in this thesis is part of the framework described by D. Rybach in this publication.

[Lehnen & Hahn$^+$ 12] The Idea for this work about hidden conditional random fields came from P. Lehnen and the first implementation was done by A. Guta. This implementation was verified by a re-implementation done by P. Lehnen and S. Hahn.

[Lehnen & Allauzen$^+$ 13] The original idea for this comparison of two conditional random field toolkits came from F. Yvon[7] and H. Ney. The experiments were done, analysed and presented by P. Lehnen, A. Allauzen[7], and T. Lavergne[7].

[Hahn & Lehnen$^+$ 13] In this publication S. Hahn integrated the hidden conditional random fields implementation in this thesis into a LVCSR system. P. Lehnen is mentioned as an author because he supported the software package and gave support in the experiments. S. Hahn had the idea to these experiments and prepared and conducted them.

[Lehnen & Peter$^+$ 13] This idea for conditional random fields applied to statistical machine translation and the modifications to the hidden conditional random fields software came from P. Lehnen. He was supported by n-best lists from J. T. Peter. The first SMT system and forced alignments were provided by J. Wübker and the second SMT system was provided by S. Peitz.

[Wuebker & Muehr$^+$ 15] J. Wübker had the idea for these experiments with maximum expected BLEU and modified the phrase model training for them. P. Lehnen proposed to change the algorithm to use RPROP, L2-regularization, and not to update the original models but to add the features as additional models. The first implementation and experiments were mainly done by S. Mühr and extended by J. Wübker.

[Lehnen & Schäpers$^+$ 07] The content of this paper is not part of this thesis. It was published as a result of the Diploma thesis in the physics department.

A. Appendix

A.1 RPROP-Elastic-Net-Extension

The parameters in the conditional random fields experiments are all optimized with the help of the resilient propagation (RPROP, [Riedmiller & Braun 93]). Each iteration m the parameters λ_1^R are updated based on the sign of the gradient times a step size $s_{r,m}$ per parameter. In Figure A.1b a typical update of a parameter λ_r is shown. The iteration starts at 0 and steps with an initial step size in the direction of the gradient. In the next step the sign of the gradient has not changed causing to increase the step size by s_+. Step 2 has crossed the root of the objective function $\frac{\partial L}{\partial \lambda_r}$ yielding the execution of the second block in the if statement of Figure A.1a. The value of λ_r is reset to the one of iteration 1 and the step size is reduced with s_-. As in step 2 the gradient $\frac{\partial L}{\partial \lambda_r}$ was memorized as zero, step 3 executes the third if block, taking the already reduced step size and step in the direction of the gradient. In step 4 the sign of the gradient is again not changed causing an increase in the step size. Step 5, step 6, and step 7 cross the root again, letting λ_r progressively converge to the root.

The RPROP algorithm is designed for continuously differentiable functions, but including an L1-regularization $c_1||\lambda_1^R||_1$ in the conditional log-likelihood L (Equation (4.5)) results in a jump in the gradient of L (curve "with L1" (dotted) in Figure A.2). The gradient can be decomposed into a part without regularization $\partial L_0/\partial \lambda_r$ and the elastic-net regularization:

$$\frac{\partial L'}{\partial \lambda_r} = \frac{\partial L}{\partial \lambda_r} - 2c_2\lambda_r - c_1 \operatorname{sign}(\lambda_r) \tag{A.1}$$

With $|\lambda_r| \gg 0$, the L1-regularization part of Equation (A.1) only changes the final optimum of λ_r by a small offset, but next to $\lambda_r = 0$ there are two distinct cases sketched in Figure A.2:

a The regular RPROP algorithm will set the final parameter $\lambda_e = 0$ after an infinite number of iterations.

b It will trim it to the final parameter $\lambda_e \neq 0$.

It would be desirable to recognize the case (a) already in one or two iterations without confusing it with case (b). As can be seen from Figure A.2, the decision between the two cases is equivalent to the decision if the gradient approaches the axis $\lambda_r = 0$ on different sides of the axis $\frac{\partial L}{\partial \lambda_r} = 0$. RPROP is an iterative procedure. Thus, a decision between case (a) and case (b) has to be done from the gradient known on different points $\lambda_{r,m}$. If the decision is wrong for one iteration the algorithm needs to recover in the next iterations.

With Equation (A.1) the jump in the gradient is

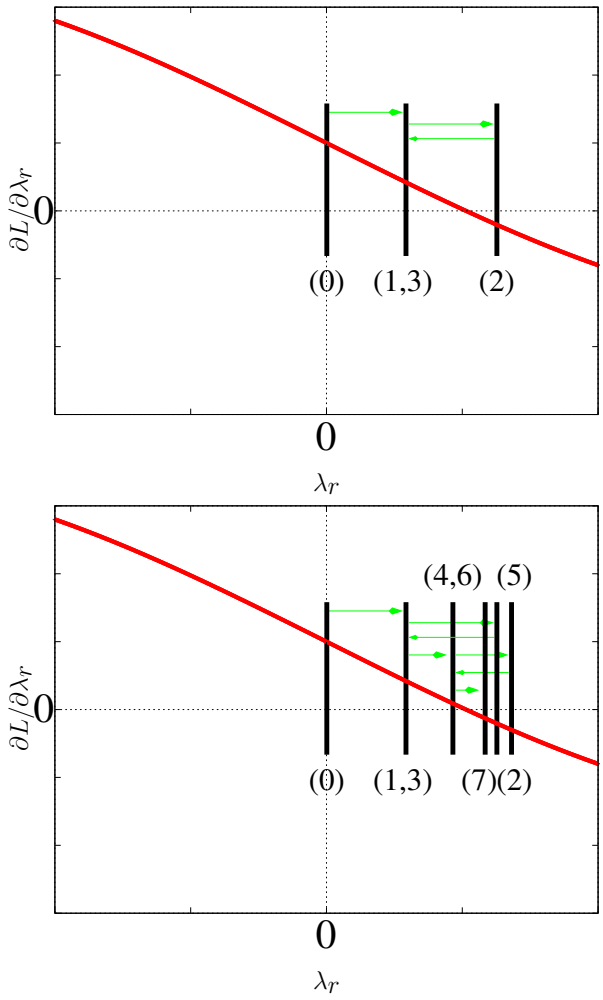
$$\lim_{\lambda_r \to -0} \frac{\partial L'}{\partial \lambda_r} - \lim_{\lambda_r \to +0} \frac{\partial L'}{\partial \lambda_r} = 2c_1.$$

Thus, the decision for a jump around $\frac{\partial L'}{\partial \lambda_r} = 0$ is given if the sign of the full gradient $\frac{\partial L'}{\partial \lambda_r}$ can be changed by the L1-regularization part $c_1 \operatorname{sign}(\lambda_r)$. In the case $\lambda_{m,r} = 0$ (e.g. in the beginning of

Input:
last and current lambdas $\{\lambda_1^R\}_{m-1}$, $\{\lambda_1^R\}_m$,
current step sizes $\{s_1^R\}_m$,
last and current gradient $\{\nabla_{\lambda_1^R} L\}_{m-1}$, $\{\nabla_{\lambda_1^R} L\}_m$
Output:
new lambdas $\{\lambda_1^R\}_{m+1}$,
new step sizes $\{s_1^R\}_{m+1}$
Options:
s_+, s_-, s_{max}, s_{min}

for $r \in 1, \ldots, R$:
  if $\{\frac{\partial L}{\partial \lambda_r}\}_{m-1} \cdot \{\frac{\partial L}{\partial \lambda_r}\}_m > 0$:
    $s_{r,m+1} = \min(s_+ \cdot s_{r,m}, s_{max})$
    $\lambda_{r,m+1} = \lambda_{r,m} - sign(\{\frac{\partial L}{\partial \lambda_r}\}_m) \cdot s_{r,m+1}$
  else if $\{\frac{\partial L}{\partial \lambda_r}\}_{m-1} \cdot \{\frac{\partial L}{\partial \lambda_r}\}_m < 0$:
    $s_{r,m+1} = \max(s_- \cdot s_{r,m}, s_{min})$
    $\lambda_{r,m+1} = \lambda_{r,m-1}$
    $\{\frac{\partial L}{\partial \lambda_r}\}_m = 0$
  else if $\{\frac{\partial L}{\partial \lambda_r}\}_{m-1} \cdot \{\frac{\partial L}{\partial \lambda_r}\}_m == 0$:
    $s_{r,m+1} = s_{r,m}$
    $\lambda_{r,m+1} = \lambda_{r,m} - sign(\{\frac{\partial L}{\partial \lambda_r}\}_m) \cdot s_{r,m}$

(a) algorithm



(b) graphical demonstration

Figure A.1 RPROP Algorithm as proposed in [Riedmiller & Braun 93]. s_+, s_-, s_{max}, s_{min} are configuration variables typically defined as $s_+ = 1.2$, $s_- = 0.5$, $s_{max} = 50$, $s_{min} = 0$.

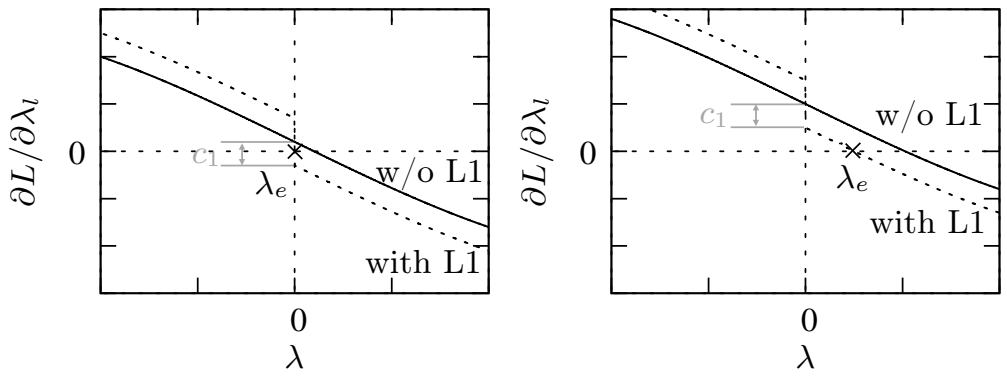


Figure A.2 Sketches of the gradient of the objective function from Equation (A.1) with equal L1 regularization c_1 but different offset.

the algorithm), the decision between case a and case b can be determined exactly with

$$\left| \frac{\partial L}{\partial \lambda_r}\Big|_{\lambda_r=\lambda_{r,m}=0} \right| = \left| \frac{\partial L'}{\partial \lambda_r}\Big|_{\lambda_r=\lambda_{r,m}=0} \right| < c_1 \tag{A.2}$$

(with $\text{sign}(0) = 0$) as the gradients are known at the jump in the gradient. While the algorithm proceeds, the value of λ_r rarely reaches exact $\lambda_r = 0$ and the rule in Equation (A.2) cannot applied again. Thus, a check for a jump in the gradient is included if the zero crossing of the gradient is passed between the iteration m and its predecessor $m - 1$:

$$\lambda_{r,m} \cdot \lambda_{r,m-1} < 0, \tag{A.3}$$

One possible solution in this situation would be to set $\lambda_{r,m} = 0$, recalculate the gradient and check the condition in Equation (A.2) in the next iteration. However, we can distinguish between three cases: (1) The gradient with L1 regularization has not changed its sign. Then the jump crosses zero, we have case (b), and setting $\lambda_{r,m} = 0$ is not needed. (2) The sign of the gradient has changed

$$\left\{ \frac{\partial L'}{\partial \lambda_r} \right\}_{m-1} \cdot \left\{ \frac{\partial L'}{\partial \lambda_r} \right\}_{m} < 0 \tag{A.4}$$

and the gradient jumps over the axis $\frac{\partial L'}{\partial \lambda_r} = 0$. Assuming λ_r to be small enough, this is given, similar to Equation (A.2), if the distance of the gradient without regularization $\partial L / \partial \lambda_r$ to the axis is smaller than c_1:

$$\begin{aligned} &\left| \frac{\partial L}{\partial \lambda_r}\Big|_{\lambda_r=\lambda_{r,m}} \right| < c_1 \\ \Leftrightarrow &\left| \frac{\partial L'}{\partial \lambda_r}\Big|_{\lambda_r=\lambda_{r,m}} + 2c_2\lambda_{r,m} + c_1 \,\text{sign}(\lambda_{r,m}) \right| < c_1. \end{aligned} \tag{A.5}$$

In this case, either the distance to the axis $\lambda_r = 0$ is too large (case $2'$) or actually the jump surrounds the axis (case $2''$). As the regular RPROP algorithm would iterate around $\lambda_r = 0$ and slowly approach $\lambda_r = 0$ in the case of a jump surrounding the axis, the $\lambda_{r,m}$ is set to zero if Equations A.3, A.4, and A.5 are fulfilled. In the next iteration, the gradient is recalculated, and the condition in Equation (A.2) is checked. If setting $\lambda_{r,m} = 0$ has been wrong (case $2'$) the condition in Equation (A.2) is not fulfilled in the next iteration and the algorithm recovers. In the last case (3) the gradient has changed its sign (Equation (A.4)) but the condition in Equation (A.5) is not fulfilled. It is likely that the gradient does not cross zero (case $3'$). However, it could be again that λ_r is too large (case $3''$). Already the regular RPROP will reduce the step size and change update direction in this case. Thus, it can be expected that the jump in the gradient is passed in the next two iterations again if the jump crosses zero. The decision can be postponed.

By application of the rules in Equations A.3, A.4, and A.5 the convergence can be speed up in case 1 and $2''$. All wrong decisions are recovered in the next one or two iterations. Like in the regular RPROP algorithm, the λ_r in Equations A.3, A.4, and A.5 are assumed as independent. Only the step size takes the other parameters $\lambda_{r' \neq r}$ into account. It could happen that in one iteration the λ_r should be set to zero, and in later iterations when the other $\lambda_{r' \neq r}$ have changed it should be $\lambda_r \neq 0$.

The final extended RPROP algorithm is presented in Figure A.3.

Input: Last and current parameters $\{\lambda_1^R\}_{m-1}$, $\{\lambda_1^R\}_m$, current step sizes $\{s_1^R\}_m$, last and current gradient of the objective function $\{\nabla_{\lambda_1^R} L\}_{m-1}$, $\{\nabla_{\lambda_1^R} L\}_m$. m enumerates the iteration.
Output: New parameters $\{\lambda_1^R\}_{m+1}$, new step sizes $\{s_1^R\}_{m+1}$
for $r \in 1, \ldots, R$:
  if $\lambda_{r,m} = 0$ and $|\{\frac{\partial L}{\partial \lambda_r}\}_m| < c_1$
    $\lambda_{r,m+1} = 0$
  else
    if $\{\frac{\partial L}{\partial \lambda_r}\}_{m-1} \cdot \{\frac{\partial L}{\partial \lambda_r}\}_m > 0$:
      $s_{r,m+1} = \min(s^+ \cdot s_{r,m}, s_{max})$
      $\lambda_{r,m+1} = \lambda_{r,m} - \text{sign}(\{\frac{\partial L}{\partial \lambda_r}\}_m) \cdot s_{r,m+1}$
    else if $\{\frac{\partial L}{\partial \lambda_r}\}_{m-1} \cdot \{\frac{\partial L}{\partial \lambda_r}\}_m < 0$:
      if $\lambda_{r,m} \cdot \lambda_{r,m-1} < 0$ *and* $\left|\{\frac{\partial L}{\partial \lambda_r}\}_m + 2c_2\lambda_{r,m} + c_1 \,\text{sign}(\lambda_{r,m})\right| < c_1$
        $\lambda_{r,m+1} = 0$
      else
        $s_{r,m+1} = \max(s^- \cdot s_{r,m}, s_{min})$
        $\lambda_{r,m+1} = \lambda_{r,m-1}$
        $\{\frac{\partial L}{\partial \lambda_r}\}_m = 0$
    else if $\{\frac{\partial L}{\partial \lambda_r}\}_{m-1} \cdot \{\frac{\partial L}{\partial \lambda_r}\}_m == 0$:
      $s_{r,m+1} = s_{r,m}$
      $\lambda_{r,m+1} = \lambda_{r,m} - \text{sign}(\{\frac{\partial L}{\partial \lambda_r}\}_m) \cdot s_{r,m}$

Figure A.3 RPROP algorithm as proposed in [Riedmiller & Braun 93] with extensions described in Section A.1 (extensions are marked with italic font, regular algorithm in gray). s^+, s^-, s_{max}, s_{min} are configuration variables typically defined as $s^+ = 1.2$, $s^- = 0.5$, $s_{max} = 50$, $s_{min} = 0$. c_1, c_2 are the L1/L2 configuration parameters defined in Equation (4.5).

List of Figures

List of Tables

Bibliography

[Allauzen & Bonneau-Maynard+ 11] A. Allauzen, H. Bonneau-Maynard, H.S. Le, A. Max, G. Wisniewski, F. Yvon, G. Adda, J.M. Crego, A. Lardilleux, T. Lavergne, A. Sokolov: LIMSI @ WMT11. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, WMT '11, pp. 309–315, Edinburgh, UK, July 2011.

[Allauzen & Mohri+ 03] C. Allauzen, M. Mohri, B. Roark: Generalized algorithms for constructing statistical language models. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1*, ACL '03, pp. 40–47, Sapporo, Japan, July 2003.

[Allauzen & Riley+ 07] C. Allauzen, M. Riley, J. Schalkwyk, W. Skut, M. Mohri: OpenFst: A General and Efficient Weighted Finite-State Transducer Library. In *Proceedings of the 12th International Conference on Implementation and Application of Automata*, CIAA'07, pp. 11–23, Praque, Czech Republic, July 2007.

[Altun & Tsochantaridis+ 03] Y. Altun, I. Tsochantaridis, T. Hofmann: Hidden markov support vector machines. In *Proceedings of the 20th International Conference on Machine Learning, ICML*, pp. 3–10, Washington, DC, USA, Aug. 2003.

[Auli & Galley+ 14] M. Auli, M. Galley, J. Gao: Large-scale Expected BLEU Training of Phrase-based Reordering Models. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1250–1260, Doha, Qatar, Oct. 2014.

[Baayen & Piepenbrock+ 95] R.H. Baayen, R. Piepenbrock, L. Gulikers: CELEX2 LDC96L14, 1995. https://catalog.ldc.upenn.edu/docs/LDC96L14/.

[Bahdanau & Cho+ 15] D. Bahdanau, K. Cho, Y. Bengio: Neural Machine Translation by Jointly Learning to Align and Translate. *Proceedings of the International Conference on Machine Translation*, Vol. abs/1409.0, may 2015.

[Baker & Fillmore+ 98] C.F. Baker, C.J. Fillmore, J.B. Lowe: The Berkeley FrameNet Project. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics*, Vol. 1 of *ACL '98*, pp. 86–90, Montreal, Quebec, Canada, Aug. 1998.

[Basha Shaik & Rybach+ 12] M.A. Basha Shaik, D. Rybach, S. Hahn, R. Schlüter, H. Ney: Hierarchical Hybrid Language models for Open Vocabulary Continuous Speech Recognition using WFST. In *Proceedings of the Workshop on Statistical and Perceptual Audition (SAPA - SCALE)*, pp. 46–51, Portland, OR, USA, Sept. 2012.

[Baumann 13] T. Baumann: *Incremental Spoken Dialogue Processing: Architecture and Lower-level Components*. Ph.D. thesis, Faculty of Linguistics and Literary Studies, Bielefeld University, Germany, 2013.

[Bender 10] O. Bender: *Robust Machine Translation for Multi-Domain Tasks*. Ph.D. thesis, Computer Science Department, RWTH Aachen University, 2010.

[Berger & Miller 98] A. Berger, R. Miller: Just-in-time language modelling. In *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing*, Vol. 2, pp. 705–708, Seattle, WA, USA, May 1998.

[Bilmes & Kirchhoff 03] J.A. Bilmes, K. Kirchhoff: Factored language models and generalized parallel backoff. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology, NAACL*, NAACL-Short '03, pp. 4–6, Edmonton, Canada, May 2003.

[Bilmes & Zweig 02] J. Bilmes, G. Zweig: The graphical models toolkit: An open source software system for speech and time-series processing. In *Proceedings of the 2002 IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP*, Vol. 4, pp. IV–3916–IV–3919, Orlando, FL, USA, May 2002.

[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.

[Bishop & Nasrabadi 06] C.M. Bishop, N.M. Nasrabadi: *Pattern recognition and machine learning*, Vol. 1. Springer New York. ISBN: 0-387-31073-8, 2006.

[Blunsom & Cohn$^+$ 08] P. Blunsom, T. Cohn, M. Osborne: A discriminative latent variable model for statistical machine translation. *Proceedings of the 46th Annual Meeting of the Association for Computational Lingusitics: Human Language Technologies, ACL-HLT*, Vol. 1, pp. 200–208, June 2008.

[Bojar & Buck$^+$ 14] O. Bojar, C. Buck, C. Federmann, B. Haddow, P. Koehn, J. Leveling, C. Monz, P. Pecina, M. Post, H. Saint-Amand, R. Soricut, L. Specia, A. Tamchyna: Findings of the 2014 workshop on statistical machine translation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation, WMT*, pp. 12–58, Baltimore, MD, USA, June 2014.

[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 *Proceedings of the Fifth International Conference on Language Resources and Evaluation, LREC*, pp. 2054–2059, Genoa, Italy, May 2006.

[Brown & Della Pietra$^+$ 93] P.F. Brown, V.J. Della Pietra, S.A. Della Pietra, R.L. Mercer: The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, Vol. 19, No. 2, pp. 263–311, 1993.

[Callison-Burch & Osborne$^+$ 06] C. Callison-Burch, M. Osborne, P. Koehn: Re-evaluating the Role of BLEU in Machine Translation Research. In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 249–256, Trento, Italy, April 2006.

[Camelin & Béchet$^+$ 10] N. Camelin, F. Béchet, G. Damnati, R. De Mori: Detection and Interpretation of Opinion Expressions in Spoken Surveys. *IEEE Transactions on Audio, Speech and Language Processing*, Vol. 18, No. 2, pp. 369–381, Feb. 2010.

[Casacuberta & Vidal 04] F. Casacuberta, E. Vidal: Machine translation with inferred stochastic finite-state transducers. *Computational Linguistics*, Vol. 30, No. 2, pp. 205–225, June 2004.

[Chen & Goodman 99] S.F. Chen, J. Goodman: An Empirical Study of Smoothing Techniques for Language Modeling. *Computer Speech & Language*, Vol. 13, No. 4, pp. 359–394, June 1999.

[Cherry 13] C. Cherry: Improved Reordering for Phrase-Based Translation using Sparse Features. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 22–31, Atlanta, Georgia, June 2013.

[Cherry & Moore$^+$ 12] C. Cherry, R.C. Moore, C. Quirk: On hierarchical re-ordering and permutation parsing for phrase-based decoding. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, WMT '12, pp. 200–209, Montreal, Quebec, Canada, June 2012.

[Chiang 05] D. Chiang: A Hierarchical Phrase-Based Model for Statistical Machine Translation. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pp. 263–270, Ann Arbor, MI, USA, June 2005.

[Chiang & Marton$^+$ 08] D. Chiang, Y. Marton, P. Resnik: Online large-margin training of syntactic and structural translation features. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pp. 224–233, Honolulu, Hawaii, Oct. 2008.

[Clark & Dyer$^+$ 11] J.H. Clark, C. Dyer, A. Lavie, N.A. Smith: Better Hypothesis Testing for Statistical Machine Translation: Controlling for Optimizer Instability. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 176–181, Portland, Oregon, USA, June 2011.

[Crammer & Dekel$^+$ 06] K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, Y. Singer: Online Passive-Aggressive Algorithms. *The Journal of Machine Learning Research*, Vol. 7, pp. 551–585, Dec. 2006.

[Crammer & Singer 03] K. Crammer, Y. Singer: Ultraconservative Online Algorithms for Multiclass Problems. *The Journal of Machine Learning Research*, Vol. 3, pp. 951–991, March 2003.

[Dahl & Bates$^+$ 94] D.A. Dahl, M. Bates, M. Brown, W. Fisher, K. Hunicke-Smith, D. Pallett, C. Pao, A. Rudnicky, E. Shriberg: Expanding the Scope of the ATIS Task: The ATIS-3 Corpus. In *Proceedings of the Workshop on Human Language Technology*, HLT '94, pp. 43–48, Plainsboro, NJ, USA, March 1994.

[Dean & Kanazawa 88] T.L. Dean, K. Kanazawa: Probabilistic Temporal Reasoning. In *Proceedings of the 7th National Conference on Artificial Intelligence*, AAAI '88, pp. 524–529, St. Paul, MN, USA, Aug. 1988.

[Deligne & Yvon$^+$ 95] S. Deligne, F. Yvon, F. Bimbot: Variable-Length Sequence Matching for Phonetic Transcription Using Joint Multigrams. In *Proceeding of the Fourth European Conference on Speech Communication and Technology*, EUROSPEECH, pp. 2243–2246, Madrid, Spain, Sept. 1995.

[Deoras & Tür$^+$ 13] A. Deoras, G. Tür, R. Sarikaya, D.Z. Hakkani-Tür: Joint Discriminative Decoding of Words and Semantic Tags for Spoken Language Understanding. *IEEE Transactions on Audio, Speech and Language Processing*, Vol. 21, No. 8, pp. 1612–1621, Aug. 2013.

[Deselaers & Hasan+ 09] T. Deselaers, S. Hasan, O. Bender, H. Ney: A Deep Learning Approach to Machine Transliteration. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, StatMT, pp. 233–241, Athens, Greece, March 2009.

[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 *Proceedings of the 2nd Workshop on Semantic Representation of Spoken Language*, SRSL, pp. 34–41, Athens, Greece, March 2009.

[Galley & Manning 08] M. Galley, C.D. Manning: A simple and effective hierarchical phrase reordering model. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP, pp. 848–856, Honolulu, Hawaii, US, Oct. 2008.

[Gao & He 13] J. Gao, X. He: Training MRF-Based Phrase Translation Models using Gradient Ascent. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, NAACL HLT, pp. 450–459, Atlanta, Georgia, USA, June 2013.

[Gimpel & Smith 10] K. Gimpel, N.A. Smith: Softmax-Margin CRFs: Training Log-Linear Models with Cost Functions. In *Proceedings of the 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 733–736. 2010.

[Goodman 01] J. Goodman: Classes for fast maximum entropy training. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, Vol. 1 of *ICASSP '01*, pp. 561–564, Salt Lake City, UT, USA, May 2001.

[Green & Wang+ 13] S. Green, S. Wang, D. Cer, C.D. Manning: Fast and Adaptive Online Training of Feature-Rich Translation Models. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, Vol. 1 of *ACL '13*, pp. 311–321, Sofia, Bulgaria, Aug. 2013.

[Gunawardana & Mahajan+ 05] A. Gunawardana, M. Mahajan, A. Acero, J.C. Platt: Hidden conditional random fields for phone classification. In *Proceedings of the 9th European Conference on Speech Communication and Technology*, INTERSPEECH-Eurospeech '05, pp. 1117–1120, Lisbon, Portugal, Sept. 2005.

[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 Transactions on Audio, Speech, and Language Processing*, Vol. 19, No. 6, pp. 1569–1583, Aug. 2011.

[Hahn & Lehnen+ 08a] S. Hahn, P. Lehnen, H. Ney, C. Raymond: System combination for spoken language understanding. In *Proceedings of the 9th Annual Conference of the International Speech Communication Association*, Interspeech '08, 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 *Proceedings of the Sixth International Conference on Language Resources and Evaluation*, LREC '08, pp. 2947–2950, Marrakech, Morocco, May 2008.

[Hahn & Lehnen+ 09] S. Hahn, P. Lehnen, G. Heigold, H. Ney: Optimizing CRFs for SLU Tasks in Various Languages Using Modified Training Criteria. In *Proceedings of 10th Annual Conference of the International Speech Communication Association*, Interspeech '09, 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 *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, ICASSP '11, 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 *Proceedings of the 14th Annual Conference of the International Speech Communication Association*, INTERSPEECH '13, pp. 495–499, Lyon, France, Aug. 2013.

[Hahn & Vozila$^+$ 12] S. Hahn, P. Vozila, M. Bisani: Comparison of Grapheme-to-Phoneme Methods on Large Pronunciation Dictionaries and LVCSR Tasks. In *Proceedings of the 13th Annual Conference of the International Speech Communication Association*, Interspeech '12, pp. 2538–2541, Portland, OR, USA, Sept. 2012.

[Hasan & Ganitkevitch$^+$ 08] S. Hasan, J. Ganitkevitch, H. Ney, J. Andrés-Ferrer: Triplet Lexicon Models for Statistical Machine Translation. In *Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, pp. 372–381, Honolulu, Hawaii, Oct. 2008.

[He & Deng 12] X. He, L. Deng: Maximum expected BLEU training of phrase and lexicon translation models. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, Vol. 1 of *ACL '12*, pp. 292–301, Jeju Island, Korea, July 2012.

[Heigold & Hahn$^+$ 11] G. Heigold, S. Hahn, P. Lehnen, H. Ney: EM-Style Optimization of Hidden Conditional Random Fields for Grapheme-to-Phoneme Conversion. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 4920–4923, Prague, Czech Republic, May 2011.

[Heigold & Lehnen$^+$ 08] G. Heigold, P. Lehnen, R. Schlüter, H. Ney, R. Schluter: On the equivalence of Gaussian and log-linear HMMs. In *Proceedings of the 9th Annual Conference of the International Speech Communication Association*, Interspeech '08, pp. 273–276, Brisbane, Australia, Sept. 2008.

[Heigold & Schlüter$^+$ 09] G. Heigold, R. Schlüter, H. Ney: Modified MPE/MMI in a Transducer-Based Framework. In *Proceedings of the {IEEE} International Conference on Acoustics, Speech, and Signal Processing*, ICASSP '09, pp. 3749–3752, Taipei, Taiwan, April 2009.

[Heigold & Wiesler$^+$ 10] G. Heigold, S. Wiesler, M. Nussbaum, P. Lehnen, R. Schlüter, H. Ney: Discriminative HMMs, Log-Linear Models, and CRFs: What is the Difference? In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 5546–5549, Dallas, Texas, USA, March 2010.

[Hemphill & Godfrey$^+$ 90] C.T. Hemphill, J.J. Godfrey, G.R. Doddington: The ATIS Spoken Language Systems Pilot Corpus. In *Proceedings of the Third DARPA Speech and Natural Language Workshop*, HLT '90, pp. 96–101, Hidden Valley, PA, USA, June 1990.

[Hopkins & May 11] M. Hopkins, J. May: Tuning as ranking. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pp. 1352–1362, Edinburgh, Scotland, July 2011.

[Huck & Ratajczak$^+$ 10] M. Huck, M. Ratajczak, P. Lehnen, H. Ney: A Comparison of Various Types of Extended Lexicon Models for Statistical Machine Translation. In *Conference of the Association for Machine Translation in the Americas 2010*, Denver, Colorado, USA, Oct. 2010.

[Huck & Wuebker+ 13] M. Huck, J. Wuebker, F. Rietig, H. Ney: A Phrase Orientation Model for Hierarchical Machine Translation. In *Workshop on Statistical Machine Translation*, WMT '13, pp. 452–463, Sofia, Bulgaria, Aug. 2013.

[Ittycheriah & Roukos 07] A. Ittycheriah, S. Roukos: Direct Translation Model 2. In *Proceedings of the North American Chapter of the Association for Computational Linguistics - Human Language Technologies conference*, NAACL-HLT, pp. 57–64, Rochester, NY, USA, April 2007.

[Jiampojamarn & Cherry+ 10] S. Jiampojamarn, C. Cherry, G. Kondrak: Integrating Joint n-gram Features into a Discriminative Training Framework. In *Proceeding of the 11th Annual Conference of the North American Chapter of the Association for Computational Linguistics*, NAACL-HLT '10, pp. 697–700, Los Angeles, CA, USA, June 2010.

[Jiampojamarn & Kondrak+ 07] S. Jiampojamarn, G. Kondrak, T. Sherif: Applying Many-to-Many Alignments and Hidden Markov Models to Letter-to-Phoneme Conversion. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, NAACL-HTL'07, pp. 372–379, Rochester, NY, USA, April 2007.

[Jiampojamarn & Kondrak 09] S. Jiampojamarn, G. Kondrak: Online Discriminative Training for Grapheme-to-Phoneme Conversion. In *Proceedings of 10th Annual Conference of the International Speech Communication Association*, pp. 1303–1306, Brighton, U.K., Sept. 2009.

[Kanthak & Ney 04] S. Kanthak, H. Ney: FSA: An Efficient and Flexible C++ Toolkit for Finite State Automata Using On-Demand Computation. In *Proc. of the 42nd Annual Meeting of the Association for Computational Linguistics*, ACL, pp. 510–517, Barcelona, Spain, July 2004.

[Kneser & Ney 93] R. Kneser, H. Ney: Forming Word Classes by Statistical Clustering for Statistical Language Modelling. In R. Köhler, B. Rieger, editors, *Contributions to Quantitative Linguistics: Proceedings of the First International Conference on Quantitative Linguistics*, QUALICO, pp. 221–226, Trier, Germany, Sept. 1993.

[Kneser & Ney 95] R. Kneser, H. Ney: Improved backing-off for m-gram language modeling. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, ICASSP '95, pp. 181–184, Detroit, MI, USA, May 1995.

[Koehn 10] P. Koehn: *Statistical machine translation.* Cambridge University Press. ISBN: 978-0-521-87415-1, 2010.

[Koehn & Och+ 03] P. Koehn, F.J. Och, D. Marcu: Statistical phrase-based translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, Vol. 1 of *NAACL '03*, pp. 48–54, Edmonton, Canada, May 2003.

[Kominek & Black 06] J. Kominek, A.W. Black: Learning Pronunciation Dictionaries: Language Complexity and Word Selection Strategies. In *Proceedings of the Main Conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, HLT-NAACL '06, pp. 232–239, New York, NY, USA, June 2006.

[Koo & Collins 05] T. Koo, M. Collins: Hidden-variable models for discriminative reranking. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT/EMNLP '05, pp. 507–514, Vancouver, British Columbia, Canada, Oct. 2005.

[Kudo 05] T. Kudo: CRF++ toolkit, 2005. https://taku910.github.io/crfpp/.

[Kudo & Matsumoto 01] T. Kudo, Y. Matsumoto: Chunking with Support Vector Machines. In *Proceedings of the second meeting of the North American Chapter of the Association for Computational Linguistics on Language technologies 2001*, NAACL '01, pp. 1–8, Pittsburgh, PA, USA, June 2001.

[Lafferty & McCallum$^+$ 01] J. Lafferty, A. McCallum, F. Pereira: Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML, pp. 282–289, Williamstown, MA, USA, June 2001.

[Lavergne & Allauzen$^+$ 11] T. Lavergne, A. Allauzen, J.M. Crego, F. Yvon: From n-gram-based to CRF-based Translation Models. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, WMT '11, pp. 542–553, Edinburgh, UK, July 2011.

[Lavergne & Cappé$^+$ 10] T. Lavergne, O. Cappé, F. Yvon: Practical Very Large Scale CRFs. In *Proceedings the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pp. 504–513, Uppsala, Sweden, July 2010.

[Le & Allauzen$^+$ 12] H.S. Le, A. Allauzen, F. Yvon: Continuous Space Translation Models with Neural Networks. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, NAACL-HLT '12, pp. 39–48, Montréal, Canada, June 2012.

[Le & Oparin$^+$ 11] H.S. Le, I. Oparin, A. Allauzen, J. Gauvain, F. Yvon: Structured Output Layer neural network language model. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, ICASSP '11, pp. 5524–5527, Prague, Czech Republic, May 2011.

[Lefèvre 07] F. Lefèvre: Dynamic Bayesian Networks and Discriminative classifiers for multi-stage semantic interpretation. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, ICASSP '07, pp. 13–16, Honolulu, HI, USA, April 2007.

[Lehnen & Allauzen$^+$ 13] P. Lehnen, A. Allauzen, T. Lavergne, F. Yvon, S. Hahn, H. Ney: Structure Learning in Hidden Conditional Random Fields for Grapheme-to-Phoneme Conversion. In *Proceedings of the 14th Annual Conference of the International Speech Communication Association*, Interspeech '13, pp. 2326–2330, Lyon, France, Aug. 2013.

[Lehnen & Hahn$^+$ 09] P. Lehnen, S. Hahn, H. Ney, A. Mykowiecka: Large-Scale Polish SLU. In *Proceedings of 10th Annual Conference of the International Speech Communication Association*, 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 *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 4916–4919, Prague, Czech Republic, May 2011.

[Lehnen & Hahn$^+$ 11b] P. Lehnen, S. Hahn, H. Ney: N-grams for Conditional Random Fields or a Failure-transition Posterior for Acyclic FSTs. In *Proceedings of the 12th Annual Conference of the International Speech Communication Association*, Interspeech '11, pp. 1437–1440, Florence, Italy, Aug. 2011.

[Lehnen & Hahn$^+$ 12] P. Lehnen, S. Hahn, A. Guta, H. Ney: Hidden Conditional Random Fields with M-to-N Alignments for Grapheme-to-Phoneme Conversion. In *Proceedings of the 13th Annual Conference of the International Speech Communication Association*, pp. 2554–2557, Portland, OR, USA, Sept. 2012.

[Lehnen & Peter$^+$ 13] P. Lehnen, J.T. Peter, J. Wuebker, S. Peitz, H. Ney: (Hidden) Conditional Random Fields Using Intermediate Classes for Statistical Machine Translation. In *Proceedings of the XIV Machine Translation Summit*, pp. 151–158, Nice, France, Sept. 2013.

[Lehnen & Schäpers$^+$ 07] P. Lehnen, T. Schäpers, N. Kaluza, N. Thillosen, H. Hardtdegen: Enhanced spin-orbit scattering length in narrow Al_ {x} Ga_ {1- x} N/ GaN wires. *Physical Review B*, Vol. 76, No. 20, pp. 205307, 2007.

[Lewis II & Stearns 68] P.M. Lewis II, R.E. Stearns: Syntax-Directed Transduction. *Journal of the Association for Computing Machinery*, Vol. 15, No. 3, pp. 465–488, July 1968.

[Li & Tur 11] X. Li, G. Tur: Spoken Query Understanding. In *Proceedings of the 12th Annual Conference of the International Speech Communication Association*, Interspeech 2011, Tutorial A4, Florence, Italy, Aug. 2011.

[Liang & Buchard-Côté$^+$ 06] P. Liang, A. Buchard-Côté, D. Klein, B. Taskar: An End-to-End Discriminative Approach to Machine Translation. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, ACL '06, pp. 761–768, Sydney, Australia, July 2006.

[Marasek & Gubrynowicz 08] K. Marasek, R. Gubrynowicz: Design and Data Collection for Spoken Polish Dialogs Database. In *Proceedings of the Sixth Int. Conf. on Language Resources and Evaluation*, LREC, pp. 185–189, Marrakech, Morocco, May 2008.

[Mariño & Banchs$^+$ 06] J.B. Mariño, R.E. Banchs, J.M. Crego, A. de Gispert, P. Lambert, J.A.R. Fonollosa, M.R. Costa-jussà: N-gram-based Machine Translation. *Journal of Computational Linguistics*, Vol. 32, No. 4, pp. 527–549, Dec. 2006.

[Mauser & Hasan$^+$ 09] A. Mauser, S. Hasan, H. Ney: Extending Statistical Machine Translation with Discriminative and Trigger-Based Lexicon Models. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, Vol. 1 of *EMNLP '09*, pp. 210–218, Singapore, Aug. 2009.

[McCallum & Freitag$^+$ 00] A. McCallum, D. Freitag, F. Pereira: Maximum entropy Markov models for information extraction and segmentation. In *Proceedings of the Seventeenth International Conference on Machine Learning*, ICML '00, pp. 591–598, Stanford, CA, USA, June 2000.

[Mohri 09] M. Mohri: Weighted automata algorithms. In M. Droste, W. Kuich, H. Vogler, editors, *Handbook of weighted automata*, pp. 213–254. Springer Berlin Heidelberg, 2009.

[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.

[Moore & Lewis 10] R.C. Moore, W. Lewis: Intelligent selection of language model training data. In *Proceedings the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pp. 220–224, Uppsala, Sweden, July 2010.

[Morin & Bengio 05] F. Morin, Y. Bengio: Hierarchical probabilistic neural network language model. In *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics*, AISTATS '05, pp. 246–252, Bridgetown, Barbados, Jan. 2005.

[Mykowiecka & Marasek$^+$ 09] A. Mykowiecka, K. Marasek, M. Marciniak, J. Rabiega-Wiśniewska, R. Gubrynowicz: Annotated Corpus of Polish Spoken Dialogues. In *Human Language Technology. Challenges of the Information Society, Third Language and Technology Conference*, Vol. 5603 of *LTC '07*, pp. 50–62, Poznan, Poland, Oct. 2009.

[Nelder & Mead 65] J. Nelder, R. Mead: A Simplex Method for Function Minimization. *The Computer Journal*, Vol. 7, pp. 308–313, 1965.

[Ney 09] H. Ney: Selected Topics in Human Language Technology and Pattern Recognition, 2009. http://www-i6.informatik.rwth-aachen.de/PostScript/Unterlagen/AdvTopics_HLT_PatRecog/SelTop_Course_Feb09.pdf.

[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 *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, Vol. 12 of *ICASSP '87*, pp. 833–836, Dallas, TX, USA, April 1987.

[NIST 95] NIST: Speech Recognition Scoring Toolkit (SCTK), 1995. http://www.nist.gov/speech/tools/.

[Nocedal & Wright 99] J. Nocedal, S.J. Wright: *Numerical Optimization*. Springer. ISBN: 978-0-387-98793-4, 1999.

[Och 95] F.J. Och: Maximum-Likelihood-Schaetzung von Wortkategorien mit Verfahren der kombinatorischen Optimierung. Studienarbeit, Universität Erlangen-Nuernberg, Germany, 1995.

[Och 03] F.J. Och: Minimum Error Rate Training for Statistical Machine Translation. In *Proceedings of the 41th Annual Meeting of the Association for Computational Linguistics*, ACL, pp. 160–167, Sapporo, Japan, July 2003.

[Och & Ney 03] F.J. Och, H. Ney: A Systematic Comparison of Various Statistical Alignment Models. *Computational Linguistics*, Vol. 29, No. 1, pp. 19–51, March 2003.

[Och & Ney 04] F.J. Och, H. Ney: The alignment template approach to statistical machine translation. *Computational linguistics*, Vol. 30, No. 4, pp. 417–449, 2004.

[Och & Tillmann+ 99] F.J. Och, C. Tillmann, H. Ney: Improved Alignment Models for Statistical Machine Translation. In *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pp. 20–28, University of Maryland, College Park, MD, USA, June 1999.

[Pal & Sutton+ 06] C. Pal, C. Sutton, A. Mccallum: Sparse forward-backward using minimum divergence beams for fast training of conditional random fields. In *Proceedings of the IEEE International Conference on Acoustics Speech and Signal Processing*, ICASSP 2006, pp. 581–584, Toulouse, France, May 2006.

[Papineni & Roukos+ 02] K.A. Papineni, S. Roukos, T. Ward, W.J.J. Zhu: Bleu: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pp. 311–318, Philadelphia, PA, USA, July 2002.

[Parihar & Picone 02] N. Parihar, J. Picone: Aurora Working Group: DSR Front End LVCSR Evaluation AU/384/02. Technical report, Institute for Signal and Information Processing, Department of Electrical and Computer Engineering, Mississippi State University, 2002.

[Peitz & Mansour+ 12] S. Peitz, S. Mansour, M. Freitag, M. Feng, M. Huck, J. Wuebker, M. Nuhn, M. Nußbaum-Thom, H. Ney: The RWTH Aachen Speech Recognition and Machine Translation System for IWSLT 2012. In *Proceedings of the International Workshop on Spoken Language Translation*, IWSLT '12, pp. 69–76, Hong Kong, Dec. 2012.

[Popović & Ney 04] M. Popović, H. Ney: Improving Word Alignment Quality using Morpho-Syntactic Information. In *Proceedings of the 20th International Conference on Computational Linguistics*, COLING '04, pp. 310–314, Geneva, Switzerland, Aug. 2004.

[Povey & Woodland 02] D. Povey, P.C. Woodland: Minimum phone error and I-smoothing for improved discriminative training. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, ICASSP '02, pp. 105–108, Orlando, FL, USA, May 2002. IEEE.

[Price 90] P.J. Price: Evaluation of spoken language systems: The ATIS domain. In *Proceedings of the Workshop on Speech and Natural Language*, HLT '90, pp. 91–95, Hidden Valley, PA, USA, June 1990.

[Quattoni & Wang+ 07] A. Quattoni, S. Wang, L.P. Morency, M. Collins, T. Darrell: Hidden Conditional Random Fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 29, No. 10, pp. 1848–1852, Oct. 2007.

[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, 2006.

[Raymond & Riccardi 07] C. Raymond, G. Riccardi: Generative and Discriminative Algorithms for Spoken Language Understanding. In *Proceedings of 8th Annual Conference of the International Speech Communication Association*, INTERSPEECH '07, 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 *Proceedings of the IEEE International Conference on Neural Networks*, Vol. 1 of *ICNN*, pp. 586–591, San Francisco, CA, USA, March 1993.

[Roark & Saraclar+ 04] B. Roark, M. Saraclar, M. Collins, M. Johnson: Discriminative language modeling with conditional random fields and the perceptron algorithm. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, number 1995 in ACL '04, pp. 47–54, Barcelona, Spain, July 2004.

[Rubenstein & Hastie+ 97] Y.D. Rubenstein, T. Hastie, Y.D. Rubinstein: Discriminative vs Informative Learning. In *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining*, KDD '97, pp. 49–53, Newport Beach, CA, USA, Aug. 1997.

[Rybach & Hahn+ 11] D. Rybach, S. Hahn, P. Lehnen, D. Nolden, M. Sundermeyer, Z. Tüske, S. Wiesler, R. Schlüter, H. Ney: RASR - The RWTH Aachen University Open Source Speech Recognition Toolkit. In *IEEE Automatic Speech Recognition and Understanding Workshop*, Hawaii, USA, Dec. 2011.

[Servan & Raymond+ 06] C. Servan, C. Raymond, F. Béchet, P. Nocéra: Conceptual Decoding from Word Lattices: Application to the Spoken Dialogue Corpus MEDIA. In *Proceedings of the Ninth International Conference on Spoken Language Processing*, INTERSPEECH-ICSLP 2006, pp. 1614–1617, Pittsburgh, PA, USA, Sept. 2006.

[Setiawan & Zhou 13] H. Setiawan, B. Zhou: Discriminative Training of 150 Million Translation Parameters and Its Application to Pruning. In *Proceedings of Conference of the North American Chapter of the Association of Computational Linguistics - Human Language Technologies*, NAACL-HLT '13, pp. 335–341, Atlanta, GA, USA, June 2013.

[Sha & Pereira 03] F. Sha, F. Pereira: Shallow parsing with conditional random fields. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, Vol. 1 of *NAACL '03*, pp. 134–141, Edmonton, Canada, May 2003.

[Shapira & Storer 02] D. Shapira, J.A. Storer: Edit Distance with Move Operations. In A. Apostolico, M. Takeda, editors, *Proceedings of the 13th Annual Symposium of Combinatorial Pattern Matching*, CPM '02, pp. 85–98. Springer Berlin Heidelberg, Fukuoka, Japan, 2373 edition, July 2002.

[Simianer & Riezler$^+$ 12] P. Simianer, S. Riezler, C. Dyer: Joint Feature Selection in Distributed Stochastic Learning for Large-scale Discriminative Training in SMT. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, Vol. 1 of *ACL '12*, pp. 11–21, Jeju Island, Korea, July 2012.

[Snover & Dorr$^+$ 06] M. Snover, B. Dorr, R. Schwartz, L. Micciulla, J. Makhoul: A Study of Translation Edit Rate with Targeted Human Annotation. In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas*, AMTA '06, pp. 223–231, Cambridge, MA, USA, Aug. 2006.

[Stolcke 02] A. Stolcke: SRILM - An Extensible Language Modeling Toolkit. In *Proceedings of the 7th International Conference on Spoken Language Processing*, ICSLP-INTERSPEECH '02, pp. 901–904, Denver, CO, USA, Sept. 2002.

[Sundermeyer & Schlüter$^+$ 12] M. Sundermeyer, R. Schlüter, H. Ney: LSTM Neural Networks for Language Modeling. In *Proceedings of the 13th Annual Conference of the International Speech Communication Association*, Interspeech '12, pp. 194–197, Portland, OR, USA, Sept. 2012.

[Sutton & McCallum 10] C. Sutton, A. McCallum: An Introduction to Conditional Random Fields. *Foundations and Trends in Machine Learning*, Vol. 4, No. 4, pp. 267–373, 2010.

[Tjong Kim Sang & Buchholz 00] E.F. Tjong Kim Sang, S. Buchholz: Introduction to the CoNLL-2000 Shared Task: Chunking. In *Proceedings of CoNLL-2000 and LLL-2000*, ConLL '00, pp. 127–132, Lisbon, Portugal, Sept. 2000.

[Tür & Mori 11] G. Tür, R.D. Mori, editors: *Spoken Language Understanding: Systems for Extracting Semantic Information from Speech.* John Wiley & Sons, Ltd. ISBN: 978-0-470-68824-3, 2011.

[Tür & Wang$^+$ 13] G. Tür, Y.Y. Wang, D.Z. Hakkani-Tür: TechWare: Spoken Language Understanding Resources [Best of the Web]. *IEEE Signal Processing Magazine*, Vol. 30, No. 3, pp. 187–189, May 2013.

[Vapnik 98] V.N. Vapnik: *Statistical learning theory.* John Wiley & Sons. ISBN: 978-0-471-03003-4, Sept. 1998.

[Vogel & Ney$^+$ 96] S. Vogel, H. Ney, C. Tillmann: HMM-Based Word Alignment in Statistical Translation. In *Proceedings of the 16th Conference on Computational Linguistics*, Vol. 2, pp. 836–841, Copenhagen, Denmark, Aug. 1996.

[Watanabe & Suzuki+ 07] T. Watanabe, J. Suzuki, H. Tsukada, H. Isozaki: Online large-margin training for statistical machine translation. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL '07, pp. 764–773, Prague, Czech Republic, June 2007.

[Woodland & Povey 02] P.C. Woodland, D. Povey: Large scale discriminative training of hidden Markov models for speech recognition. *Computer Speech & Language*, Vol. 16, No. 1, pp. 25–47, Jan. 2002.

[Wuebker & Huck+ 12] J. Wuebker, M. Huck, S. Peitz, M. Nuhn, M. Freitag, J.T. Peter, S. Mansour, H. Ney: Jane 2: Open Source Phrase-based and Hierarchical Statistical Machine Translation. In *Proceedings of the 24th International Conference on Computational Linguistics*, COLING '12, pp. 483–491, Mumbai, India, Dec. 2012.

[Wuebker & Mauser+ 10] J. Wuebker, A. Mauser, H. Ney: Training Phrase Translation Models with Leaving-One-Out. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pp. 475–484, Uppsala, Sweden, July 2010.

[Wuebker & Muehr+ 15] J. Wuebker, S. Muehr, P. Lehnen, S. Peitz, H. Ney: A Comparison of Update Strategies for Large-Scale Maximum Expected BLEU Training. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 1516–1526, Denver, CO, USA, May 2015.

[Wuebker & Peitz+ 13] J. Wuebker, S. Peitz, F. Rietig, H. Ney: Improving Statistical Machine Translation with Word Class Models. In *Conference on Empirical Methods in Natural Language Processing*, EMNLP '13, pp. 1377–1381, Seattle, USA, Oct. 2013.

[Yu & Lam 08] X. Yu, W. Lam: Hidden Dynamic Probabilistic Models for Labeling Sequence Data. In *Proceedings of the 23rd AAAI Conference on Artificial Intelligence*, pp. 739–745, Chicago, IL, USA, July 2008.

[Zens & Och+ 02] R. Zens, F.J. Och, H. Ney: Phrase-Based Statistical Machine Translation. In M. Jarke, G. Lakemeyer, J. Koehler, editors, *German Conference on Artificial Intelligence*, pp. 18–32, Aachen, Germany, Sept. 2002.

[Zhang & Jin+ 03] J. Zhang, R. Jin, Y. Yang, A.G. Hauptmann: Modified logistic regression: An approximation to svm and its applications in large-scale text categorization. In *Proceedings of the Twentieth International Conference on Machine Learning*, ICML '03, pp. 888–895, Washington, DC, USA, Aug. 2003.

[Zou & Hastie 05] H. Zou, T. Hastie: Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, Vol. 67, No. 2, pp. 301–320, April 2005.

[Zweig & Nguyen 09] G. Zweig, P. Nguyen: A segmental CRF approach to large vocabulary continuous speech recognition. In *IEEE Workshop on Automatic Speech Recognition and Understanding*, ASRU '09, pp. 152–157, Merano/Meran, Italy, Dec. 2009.